izdaje BIGZ OOUR „Duga"

računari

specijalno izdanje časopisa „Galaksija"
izlazi jedanput mesečno
cena 300 din
maj 1986

15



uputstvo za upotrebu

profi asembler 64

periferijska oprema

novi matični štampači

ekskluzivno

svi atarijevi program

kad se zakon uspava

programe bez zaštite

umetak na 32 strane

dejan ristanović

mašinac za početnike

časopis za prave pro

15 računari

Izlazi jednom mesečno · Izdaje BIGZ OOUR Duga

Specijalno izdanje časopisa „Galaksija"

Cena 300 dinara / maj 1986.

Izdaje
Beogradski izdavačko-grafički zavod
OOUR Novinska delatnost „Duga"
11000 Beograd
Bulevar vojvode Mišića 17

Telefoni
650-161 (redakcija)
650-528 (prodaja)
651-793 (propaganda)

Generalni direktor
Dobroslav Petrović

Direktor OOUR „Duga"
Bratoljub Babić

Glavni i odgovorni urednik
Gavrilo Vučković

Urednik izdanja
Jova Regasek

Tehnički urednik
Mirko Popov

Redakcija časopisa „Galaksija"
Tanasije Gavranović, pomoćnik glavnog i odgovornog urednika, Esad Jakupović, zamenik glavnog i odgovornog urednika, Aleksandar Milinković urednik, Jova Regasek urednik, Zorka Simović sekretar redakcije, Srđan Stojančev novinar, Gavrilo Vučković, glavni i odgovorni urednik

Stručna saradnja
Dejan Ristanović
Dušan Slavić
Nevenka Spalević
Anđelko Zgorelec

Stalni saradnici
Nada Aleksić, Ninoslav Čabrić, Branko Đaković, Donat Greber, Voja Gašić, Branko Hebrang, Đorđe Janković, Mihajlo Karapandžić, Vladimir Koštić, Vladimir Krstanošić, Saša D Kovačević, Srđan Kosovac, Radomir A. Mihajlović, Zvonimir Makovec, Blažimir Miše, Dejan Muhamedagić, Ivan Nador, Radomir Nikolajev, Zoran Obradović, Miodrag Potkonjak, Dejan Ristanović, Jelena Rupnik, Dušan Slavić, Jovan Skuljan, Nevenka Spalević, Srđan Stakić, Zvonimir Vistrička, Anđelko Zgorelec, Zoran Životić

Fotografije
Vladimir Simović
Ilustracije
Miodrag Marković

Izdavački savet „Galaksije"
Dr Rudi Debijadi, prof dr Branislav Dimitrijević (predsednik), Radovan Drašković, Tanasije Gavranović, Živorad Glišić, Esad Jakupović, Velizar Maslać, Nikola Pajić, Željko Perunović, prof dr Momčilo Ristić, Vlada Ristić, dr inž Miloraд Teofilović, Vidojko Veličković, Velimir Vesević, Milivoje Vuković

Štampa
Beogradski izdavačko-grafički zavod
11000 Beograd, Bulevar vojvode Mišića 17
Žiro-račun kod SDK 60802-633-2463
Devizni račun kod Beobanke 60811-620-6-62701-999-01066
Za inostranstvo cena dvostruka (400 D, 2,50 US$, 6,50 DM, 45 Sch, 5,50 Sfrs, 20 Ffrs)
Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i Službenog glasnika broj 26/72, ovo izdanje oslobođeno je poreza na promet

sadržaj



Šta ima novo

Računari na granici

Početkom maja stupili su na snagu novi carinski propisi za uvoz kućnih računara. Prilikom povratka iz inostranstva sada se može uvesti kućni kompjuter u vrednosti od 90 000 dinara. U okviru ovog iznosa mogu se uneti i tzv. „posebne računarske jedinice" — periferijska oprema čiji je uvoz do sada bio veoma sporan. Poštom se mogu primati predmeti od 30 000 dinara.

Vrednost računara se odračunava prema tzv. statističkom deviznom kursu koji se utvrđuje na početku godine i važi čitavih dvanaest meseci. Kurs za značajnije valute dat je u tabeli. Iz ovih odnosa lako je izračunati da je dozvoljen uvoz svih računara koji su jeftiniji od 340,2 američka dolara, ili 1 001,7 zapadnonemačkih maraka, ili 833,7 švajcarskih franaka, ili 650.269 italijanskih lira ili engleskih funti. Po ovom kursu odračunava se samo pravo na uvoz — carinske i ostale dažbine plaćaju se prema tekućem dnevnom deviznom kursu.

Statistički kurs

američki dolar	=	264,53 din
zapadnonemačka marka	=	89,84 din
švajcarski franak	=	107,95 din
funta sterlinga	=	343,30 din
austrijski šiling	=	12,77 din
francuski franak	=	29,42 din.
italijanska lira	=	0,13 din

Iako još uvek prilično skromno odmeravaju jugoslovenske potrebe za kućnim računarima, novi propisi ipak obezbeđuju znatno povoljnije uslove od onih koji su važili do pre neki dan. Novi limit obuhvata kompletnu klasu 128K mašina, kompletnu MSX klasu, „komodor 128 D", najnoviju klasu NLQ matričnih štampača, najbolje disk-jedinice, najkvalitetnije crno-bele monitore, a bračni par može po pola da uveze u delovima čak i nekog od IBM PC/XT klonova. Novi carinski propisi, dakle, više ne osuđuju Jugoslovene na „spektrum" i druge mašine igračke. Od dobrih kompjutera i njihove korisne primene sada nas još jedino deli sve tragičniji YU standard.

„Elektron" je Ljubišin

U nagradnoj igri „Elektron je vaš", koju je specijalno za čitaoce „Računara" organizovao „Partizan" iz Čačka, zastupnik firme Acorn za Jugoslaviju, najviše sreće imao je ***Ljubiša Petrović*** *iz* ***Niša, Nikole Kopernika 13.*** *Ljubiša je, razume se, znao da su osnivači firme Acorn Hauzed i Kari, da procesor 65C12 ima više mašinskih instrukcija od procesora 6502 i da se bočni RAM kod računara iz familije BBC najviše koristi za upisivanje sadržaja aplikativnih programa koji se obično prodaju u ROM-u, pa je na tri umereno laka pitanja Dejana Ristanovića odgovorio sa b), c) i a).*

Ljubišu je iz gomile od nekih 2000 pisama i dopisnica izvukla Jelena Rupnik (ruke joj se pozlatile!) na prvoj sednici spoljne redakcije „Računara". Jelena je odbila da se slika čak i u ovoj prilici, pa čak i iza dopisnice, pa ovu lepu vest nismo mogli da propratimo i prigodnom fotografijom. Naš foto-reporter Vladimir Simović je, srećom, među devojkama koje je ovo varljivo proleće izmamilo na ulice, pronašao i jednog ljupkog ljubitelja Acorn računara i zamolio ga za jedan studijski snimak. Ljubiši Petroviću, dakle, „elektron" sa štampačem „šinve CP80", a ostalim znalcima BBC računara i učesnicima nagradne igre jedan pomalo zagonetan osmeh naše anonimne čitateljke.



Kompjuterski poliglota

Kada smo objavili vest o ruskom tekst procesoru, javilo se nekoliko zainteresovanih kojima je trebala adresa. Evo nečeg još zanimljivijeg: poliglotski tekst procesor koji radi na engleskom, ruskom, grčkom, hebrejskom i arapskom. Zove se „Multi Lingual Scribe" i može se dobiti od firme Gamma Productions, 710 Wilshire Blvd, suite 609, Santa Monica, California, USA. Da li ste primetili da su se u istom programu našli i hebrejski i arapski? Pravi internacionalizam! (*B Đ*)

Jedna vrata zaštite

Večiti problem zaštite kompjuterskih sistema je toliko poznat da nema potrebe išta pričati o tome. Motorola je proizvela zaštitni uređaj za kompjuterske sisteme koji daje novu dimenziju zaštiti. Uređaj prima poziv, traži šifru i kada je dobije — prekida vezu. Tada u svojoj internoj telefonskoj beležnici traži telefonski broj koji ide uz šifru naziva korisnika koji se ponovo predstavlja i tek se onda uspostavlja direktna veza sa računarom. Uređaj, osim ovoga, može da kontroliše i dužinu pojedinačnog rada nekog korisnika po unapred utvrđenom rasporedu. Osnovna verzija uređaja može da komunicira sa ukupno 400 korisnika.
(*B Đ*)

Spoljna redakcija

Od broja koji upravo držite u rukama redakciji „Galaksije" u pripremanju specijalnog izdanja „Računari" pomaže i spoljna redakcija, mahom stručnih saradnika: Branko Đaković, Dejan Ristanović, Jelena Rupnik, Jovan Skuljan, prof. dr Dušan Slavić, Nevenka Spalević i Zoran Životić.

Nije sve tako crno

U poslednje vreme računari sve lakše nalaze put do naših škola. Nedavno je beogradskim školama kupljeno 100 kompjutera marke „orik nova", od ranije postoji dosta „lola" i „galaksija", a neke škole su se opremile i nekim većim mašinama. Ali u isto vreme sve češće se čuju glasovi da se kompjuteri ne koriste na pravi način: da stoje zaključani, da se njima igraju deca profesora matematike, da učenici i ne znaju za njihovo postojanje.

Beogradska škola za usmereno obrazovanje „Beogradski bataljon", poznatija kao 13. gimnazija, predstavlja sasvim suprotan primer. U računskom centru ove škole glavna zvezda je „amstrad 664" sa „epsonovim" štampačem, a pored njega tu su još jedan „spektrum", dve „lole" i dva „orika". U školi redovno radi kompjuterska sekcija, na kojoj svi učenici mogu da rade sa svim računarima, ili da nešto nauče od profesora i starijih kolega. Voleli bismo kada bi i ostale škole sledile ovaj primer i napravile svoje sekcije u saradnji sa profesorima, roditeljima i bivšim učenicima.

Drugi lep primer iz „Beogradskog bataljona" je primena računara, osim u obrazovanju, i u svakodnevnom životu i radu škole. U svoje slobodno vreme i na profesionalnoj praksi, učenici četvrte godine, u saradnji sa profesorom Miloradom Jovanovićem, napravili su nekoliko odličnih programa za školu, pomoću kojih se sređuju dnevnici, ocene, izostanci i sve što zahteva pedagoška administracija. Ovim je čitav postupak sređivanja dnevnika na kraju minulog polugodišta bio znatno olakšan i skraćen. Neka odeljenja su već dobila ocene sređene na računaru i odštampane na printeru. Osim ovog, napravljeni su programi za knjigovodstvo i proračun ličnih dohodaka. Primer „Beogradskog bataljona", verovatno, ne može bitno da ulepša inače dosta tužnu sliku u našem školstvu, ali ipak predstavlja veliko ohrabrenje.

V K



Šta ima novo

Orao dobija krila

Varaždinski PEL nedavno je završio razvoj novog školskog računara, u koji je uloženo 15 milijardi dinara Novi računar izrasteo iz galeba i orla dakle na procesoru 6502 Ima profesionalnu tastaturu, profesionalni format ekrana (tekst 80×24 grafika 620×256 tačaka) 32 K ROM-a 64 K RAM-a sa mogućnošću proširenja do 256 K disk kontroler i dve diskatne jedinice, a uskoro će biti opremljen i modulom za povezivanje u lokalnu mrežu Računar je praktično prihvačen kao standard za osnovne škole u Hrvatskoj a preduzimljivi varaždinci smatraju da ima izvesnih šansi da bude uveden i u Srbiji i već se uveliko pripremaju da otvora jedan pogon u Beogradu Za dalji razvoj ovog računara u Hrvatskoj je odvojano 60 milijardi dinara — pola od toga za hardver a pola za softver — a sredstva će biti raspoređena preko jednog opštejugoslovenskog konkursa Nemamo ništa protiv da bilo ko bez ikakvih posebnih uslova proizvodi naš računar pod bilo kojim imanom ' kažu u PEL-u i spremni smo da prihvatimo bilo čije rešenje kao svoje sopstveno ako je — bolje!



Flopi reklama

Olivetl ja pronašao način da poboljše reklamu svojih proizvoda Reklama će biti upakovana na flopi diskove i to tako što će se pod Olivetijevom oznakom prodavati diskete koja će na sebi imati demonstraciona programa a neće biti ništa skuplje od običnih diskata istog kvaliteta Pakat od deset upakovanih diskete sadrži pravu gomilu korisnog softvera uključujući i tekst procesor bazu podataka, supercalc program i slična poslastice Ako ova akcija bude imala uspeha izbor poklon softvera će verovatno biti proširen To nas raduje jel te? (*B Đ*)



Animator 1

*Srećni vlasnici **Artista i Art Studia** čuvajte se! Na tržištu se pojavio novi program za crtanje koji obuhvata i sprajt-dizajner i program za animaciju Ime mu je Animator 1 i proizvod je malo poznate softverska firme SoftCat Samo crtanje je vrlo olakšano a funkcije sa prozorima su izvanredno brze (**Art Studio**, stidi se!) Recimo, kod standardnih programa vi postavljate centar kruga a zatim jednu tačku na obodu Kod **Animatora 1** vam je sve vreme pred očima kostasti kurzor, koji se širi kako povećavate poluprečnik kruga, i tako pokazuje njegove buduće dimenzije*

Jedanput nacrtanu sliku možete i animirati pretvaranjem pojedinih njenih delova u sprajtova Inače sam sprajt-dizajner je izuzetno dodar i jednostavan za korišćenje

Sprajtovi se mogu snimiti na traku ili drajv i koristiti u bejziku ili mašinskim programima, što je sve detaljno objašnjeno u uputstvima za upotrebu

*Pliva heker preko Save nosi sprajtić na vrh glave! (**D.S.**)*

Dolazi „amiga"

Amiga se najzed pojavila u Evropi Možete je nabaviti u svim većim prodavnicama računara, a cena je onakva kakva je i bila obećana Pojavilo se i nešto malo (nekoliko stotina) programa, što predstavlja dobar početak Jedan deo programa je prerađen sa drugih računara ali ima i dosta originalnih Ohrabruje to što su se tog posla prihvatile sve veće softverske firme. Electronic Arts, jedna od najpoznatijih softverskih firmi sada radi isključivo programa za „amigu", a pored nje tu su još i Activision, Synapse Broderbund Sublogic, Lattice Mindscape Metacomco i ostali dok se za grafiku brine Islande Graphics. Pomenimo i nekoliko najinteresantnijih programa C «BASE III, „Turbo Pascal" (specijalne „amiga" verzija), „ABasic" (interpreter i kompajler), lisp makroassembler IBM emulator, „Unicalc" spredšit „Textkraft" i još dosta toga Pored ozbiljnijih programa, već su u prodaji programi za muziku crtanje, kuvanje i, naravno, igranje! Prerađene su neke starije čuvene igre, ali postoje i mnoge nove, pisane upravo za „amigu" Ove potonje predstavljaju novi prodor u video igrama Slike iz igara su prosto neverovatne za sve one koji imaju male kućne računare.

Tajnost niko ne garantuje

Ako ste mislili da niko ne može da sazna šta vi zatvorani u četiri zida radite na svom ličnom kompjuteru grdno ste se prevarili! Jedan holandski stručnjak usavršio je metod pomoću koga se iz blizine vaše zgrade može videti šta piše na ekranu vašeg monitora ili TV aparata Potrebna je samo jedna direkciona antena i elektronske oprema u vrednosti od oko 15 dolara

Igrači(ce), pakmana i ostalih muškerčina o kojima smo pisali u prošlom broju, oprez! (*B Đ*)

Laserska jabuka

Oduvek ste želeli da imate 'epl IIe ' ili IIc ili bilo koji epl ? Oduvek niste imali para za epl' ? Uobičejana stvar Zato sada možete kupiti laser' Laser 128 je mašina na kojoj rade , skoro svi programi napisani za "epl IIe III IIc. Košta samo 450 dolara i može se nabaviti kod Video Technology firme na 2633 Greenleaf Avenue, Elk Grove Village, Illinois USA Probajte Možda rade baš „skoro svi' programi (B Đ)

Laser Basic

Laser Basic *je nova varijanta jezika za „spektrum' okrenuta prvenstveno grafici i sprajtovima Program zauzima oko 8 K na vrhu RAM-a i dodaje standardnom „spektrumovom bejziku 138 novih komandi Većina služi za kontrolu grafike, mada ima i vrlo korisnih — kao što su RENUMBER i TRACE rutine*

Sprajtovi su standardne veličine 15×15 tačkice ali mogu biti i veći Unutar programa postoji i „skladišta već gotovih sprajtova, spremnih za upotrebu ima ih 109 od malih svemirskih brodova do velikih parnih lokomotive

*Naravno vaši programi obogaćeni grafikom iz „**Laser Basica**" će raditi samo kada je on prisutan u memoriji Zato su njegovi pisci izbacili na tržišta Laser Compiler (posebno se plaća) koji jednom za svagda prevodi vaše programa u mašinac.*



*Programerima koji su želeli da pišu svoje igre, a **White Lighting** im je bio suviše komplikovan **Laser Basic** će pomoći svojom jednostavnošću i efikasnošću*

*To piše i u sićušnom uputstvu od 75 stranica, koje naši pirati neće kopirati (**D.S.**)*

Opus Discovery 1

Mnoge će iznenaditi kojom brzinom su pojedini proizvođači prateće opreme reagovali na novi „spektrum 128". To se najpre može pripisati činjenici da promene u samom spektrumu nisu korenite. Treba se samo prilagoditi dodatnom ROM-u i drugoj memorijskoj banci i — to je sve.

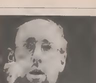
Firma Opus je poznata kao proizvođač najkompletnijeg periferijskog dodatka za spektrum. To je 3.5 inčni disk drajv Discovery koji u sebi sadrži još i paralelni interfejs za štampač, džojstik port, monitorski izlaz, konektor opšte namene i napajanje koje ujedno snabdeva i spektrum.

Priključen na „običan" „spektrum" Discovery je otvarao mogućnost formiranja RAM-diska od gornjih 32 K njegove memorije, koji je tretiran kao drajv pet. Novi Discovery priključen na „spektrum 128" utrostručuje ovu mogućnost jer omogućava da gornju memorijsku banku (64 K) koriste kao drugi RAM-disk, koji će se tretirati kao drajv šest.

Spektrum 128 poseduje jednu karakteristiku kojom baš ne može da se podiči: prelaskom u mod 128 iz moda 48 i obrnuto briše se kompletan sadržaj memorije. Međutim, Discovery i to rešava. On čak omogućava i razmenu podataka između „manjeg" i „većeg" RAM-diska. Na žalost, ovako lepe mogućnosti moraju debelo da se plate. Opus Discovery 1 sa jednom jednostranim disk-drajvom kapaciteta 178 K košta 199.95 funti, a Discovery 2 sa dva drajva i ukupnim kapacitetom od 360 K 329,95 funti.

Sic transit gloria mundi. (D.S.)

Kvantni skok u nepovrat



Početkom aprila srušeno je još jedno veliko računarsko carstvo — kompanija Sinclair Research prešla je u ruke Alana Šugera, ingenioznog šefa firme Amstrad. Šuger je odmah najavio obustavu proizvodnje kontroverznog QL-a i još kontroverznijeg mikrodrajva. Vlasnici „spektruma" ne moraju da brinu, „duga" će se i dalje proizvoditi kao mašina za — igranje. Iako je kupce svojih kompjutera često umeo da zavija u crno, Klajv Sinklar je odigrao jednu od ključnih uloga u računarskoj revoluciji. On je lansirao ideju o jeftinom računaru i pripremio put svima onima koji su ga ubrzo na njemu — pretekli!

Novi Tasword

Nedavno se u prodaji pojavila nova verzija starog dobrog tekst-procesora za „spektrum" *Tasword Two*. Nova verzija se zove, naravno, *Tasword Three (Tasword 3)* i isključivo je orijentisana na rad sa mikrodrajvom.

Pored svih standardnih funkcija koje ima i Tasword Two i brojnih dodataka i poboljšanja, novi tekst-procesor je, najzad, dobio i opciju koja mu je najviše nedostajala — MERGE za poštu. Ovoj mogućnosti će se obradovati oni koji često šalju veliki broj pisama sa istom sadržinom na različite adrese. Uz pomoć novog *Taswordа*, u procesor će samo ubaciti potrebne adrese i mogu da — odu na kafu!

Novosti su još i potpuni insert mod (dosad je, valjda, bio delimičan), štampanje do 128 znaka u redu na printeru i bafer za tastaturu koji će omogućiti brže kucanje (ne po gumicama).

Cena — prava sitnica: samo 18.90 funti. (D.S.)

Od Amstrada do PCa

Procesorsko proširenje zasnovano na 8088 koje omogućuje da na amstradu 6128 koristi softver namenjen IBM PC-ju sasvim je lep novitet za „amstradovce" (i za siromašne Pcjevce). Proširenje košta 229 funti, a to zaista nije puno za takvu mogućnost. (*B.Đ.*)

Više za mase nego za male mase

Amstrad se veoma striktno drži svoje filozofije. Dok je jeftin, dotle je i zanimljiv. Naravno, tu su i nova mašina. Jeftino, naravno. Posle PCW 8256 pojavio se i PCW 8512. PCW8256 su neki napadali, mnogi mu se čudili, a izuzetno mnogo njih ga je kupilo. Sa 8512 priča će verovatno biti nekako ista takva, osim što će oni prvi još više napadati, oni drugi mu se još više čuditi, a oni treći ga još mnogo više kupovati. Novi PCW računar namenjen obradi teksta kao i 8256 ima i printer, i monitor i disk drajv, ali ima i, uz još jedan drajv, i dvostruko više memorije — 512 kilobajta. Cena je 499 funti bez britanskog poreza na promet, koji ne plaćate ako računar iznosite iz zemlje. Pravi biser za nekog prevodioca, novinara ili saradnika „Računara".

IBM opet prvi

IBM je proizveo čitavu novu klasu računara zasnovanih na novom 32-bitnom mikroprocesoru poznatom pod imenom RISC (kompjuter sa redukovanim setom instrukcija). Osnove te klase računara je IBM RT. RT je samo dvostruko skuplji od modela AT ali je mnogo moćniji. Namenjen je CAD primenama u nauci i inženjerstvu. Novi procesor ima samo 40 za razliku od nekoliko stotina instrukcija kojima raspolažu procesori na velikim IBM-ovim računarima.

Operativni sistem RT-a je zasnovan na Unixu i zove se AIX. Računar je opremljen jednim hard diskom i jednim flopi disk drajvom.

Ovim svojim potezom IBM je pretekao Ejkorn koji je još prošle godine razvio svoj 32-bitni RISC procesor pod imenom ARM (Acorn RISC Machine), ali do sada nije uspeo da izbaci računar zasnovan na tom procesoru.

Smatra se da će ulazak IBM-a na tržište RISC procesora oživeti tržište i prokrčiti put za mnogo moćnije i mnogo jeftinije računare u bliskoj budućnosti.

Tvrda ploča

Utični hard disk (ili hard pločica) stigao je i do Evrope. Posle intenzivne reklamne kampanje u SAD i dobre početne prodaje, prvi primerci obreli su se i u Velikoj Britaniji. Postoji varijanta za IBM PC (nije za čuđenje, zar ne?) koja košta samo 775 funti. Ko voli... (*B.Đ.*)

Atarijeva PC maska

Atari je uvek bio jak u najavama. Da se ostvarilo sve što su oni najavili, ko zna gde bismo sada bili. Najnovija najava je sitna ali značajna. Atari je najavio da će uskoro jedna firma (znači ne Atari) izbaciti na tržište hardverski emulator koji ST-a pretvara u PC. Dodatak će koštati oko 300 funti i imaće ugrađen disk drajv od 5.25 inča. Navodno, moći će da se nabavi već ovog leta. To Atariju nije bilo dosta. Za malo dalju budućnost on najavljuje sličan emulator koji bi ST pretvarao u „mek". Cena se ne pominje. Zlobnici (često ih pominjemo, zar ne) pitaju se zar Atariju ne bi bilo jednostavnije da malo jače podrži TOS i da radi na programima za sam ST. Ko zna, možda su čak i u pravu.

IMB za siromašne

Sećate li se šuškanja o PC-ju II koje s vremena na vreme prostruje tržištem — poslednji put kod lansiranja PC JX-a u Australiji? Ponovo se intenzivno govori o novom IBM-ovom računaru koji bi odgovorio za pojačano interesovanje za sve one klonirane PC-je koji se proizvode na Dalekom istoku. Nova mašina bi trebalo da ima drajvove od tri i po inča i neverovatnu cenu od oko 400 funti. Strpimo se malo s tajvancima. Skupljajmo pare.

Miševi u stroj!

U Velikoj Britaniji se pojavio još jedan miš za „spektrum". Ovoga puta dolazi od slavne firme Kempston, koja njime pokušava da opravda svoj stari (hardverski) renome.

Sam miš je kvalitetnije izrađen od onoga koji proizvodi firma AMX i ima dva nezavisna tastera na sebi. Interfejs je jednostavan — ne sadrži izlaz za printer kao AMX-ov — i ne omogućuje da se programima dodaju bejzik komande za kontrolu miša. Jedino su data uputstva za očitavanje porta na koji je miš priključen.

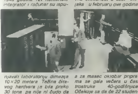
Ali Kempston ne bi bio ono što jeste da iznenađenja ne dolaze na kraju: uz pomenuti hardver dobijate i poznati izvanredni program za crtanje Art Studio, naravno prilagođen za rad sa miš(onjom). Hej, MEK, kako zdravlje?! (D.S.)



Razglednica iz Njujorka

Četrdeset godina prvog elektronskog računara

Enijak prvi veliki digitalni računar generalne primene sagrađen kompletno od elektronskih komponenti sastavljen je u Murovoj školi inženjerstva pri Univerzitetu Pensilvanija u Filadelfiji 1946 godine Tzv numerički integrator i računar su ispunjavali laboratoriju dimenzija 10×20 metara Težina čitavog hardvera je bila preko 30 tona pa nije ni čudo da je upravo engleska reč hardware (gvožđurija) bila izabrana kao opisno ime za računarski sistem bez programa Enijak je imao 70 000 otpornika 10 000 kondenzatora i 18 000 elektronskih cevi kao prekidačkih elemenata Pri izgradnji računara bilo je potrebno zalemiti preko 500 000 spojeva



Projekat Enijak je započeo u tajnosti ratne 1943 godine sa namenom da se omogući automatsko izračunavanje tabela artiljeriskih parametara za precizno gađanje Na žalost ili na sreću, enijak nije kompletiran na vreme da bi poslužio u prvobitne svrhe Da ironija bude veća prvi problem rešavan na enijaku je bio sistem diferencijalnih jednačina kojima je bila modelirana hidrogenska bomba Za nepuna dva sata enijak je bio u stanju da kompletira računarski posao ekvivalentan radu deset matematičara sa punim radnim vremenom u trajanju od deset godina

Delovi praoca svih današnjih računara se danas nalaze izloženi u Smitsonovom institutu u Vašingtonu Vojnoj akademiji Vest Point i na samom Univerzitetu Pensilvanija

Ove godine Univerzitet Pensilvanija slavi 40-godišnjicu nastanka računara enijak 40-godišnjicu prvog predmeta iz oblasti računarskih nauka koji je ikada predavan na jednom univerzitetu kao i 40-godišnjicu formiranja prve firme za proizvodnju računara Prvu računarsku kompaniju su osnovali nosioci projekta enijak — Džona Mokli (John Mauchly) i Presper Ekert (Presper J Eckert) koji su zajedno radili na enijaku sa Džonom Brajnardom (John Brainerd)

Murova škola inženjerstva je održala komemorativnu svečanost u čast enijaka u februaru ove godine a za mesec oktobar priprema se gala večera u čast trostruke 40-godišnjice Očekuje se da će 22 studenta iz prvog razreda koji je ikada slušao jedan računarski predmet (toliko ih je još u životu) biti prisutno na večeri Mnogi od studenata iz te generacije su danas poznati eksperti političari i menadžeri Dekan Murove škole dr Džozef Bordonja (Joseph Bordogna) kaže da se ta generacija od rastanka 1946 godine pa do ove godine nije sastajala

Interesantna je činjenica da današnji mikroprocesori dimenzija santimetar kvadratni sa cenom od desetak dolara imaju performanse koje prevazilaze miliondolarski enijak' iz četrdeset šeste. U jubilarno vreme kao što je ova godina dok poredimo sadašnju situaciju u računarskoj industriji sa njenim počecima pitamo se . kakvim ćemo računarskim kapacitetima raspolagati kroz nekoliko godina?

R A Mihajlović



I oad dragi računari

Svi su hakeri braća

Hej Redakcijo Jeste li našli maloj Andrei dvicu ? Niste?! Ja ih slučajno imam dvije! Pod parolom imaš puno Računara — daj malo javljam se na njeno toplo pismo i nudim joj svoju pomoć

Doduše nijedna dvica nije kao iz štamparije ali što se tu može? Naime ćoškovi su malo isfucani (zar da se osećam krivim što svaki broj okrenem desat puta u toku dana), ali su svi listovi na broju i neoštećeni

Andrea da li te interesuje ovaj predlog?

Tomislave teši se da si sada prvi na listi čekanje (Žao mi je što nemam dvicu i za tebe)

Zrnić Željko
Fruškogorska 19
78000 Banjaluka

P S Andrea, hoćeš li i meni poslati pusu?
P P S Predlažem i drugim ljubiteljima Računara koji ima više retkih brojeva da pomognu i drugim Andreama i Tomislavima Jer svi su hakeri braća! Hmm da nije ovo uzviknuo Aristotel?

Malokalibarski računar za dame

Neću da trošim puno prostora pokazujući koko čitam računare od prvog broja i da ih puno volim Preći ću odmah na naslovne strane Meni se puno sviđaju vaše naslovne strane — da ne pominjem ženske koje su na njima To je jedna činjenica Druga činjenica je to da te ženske polako istiskuju kompjutere sa naslovnih strana U poslednjem broju Računarima 14 imate jednu jako zanimljivu devojku na naslovnoj strani ali je zato računar gotovo nestao Ostalo je nešto pod imenom Casio

Nemojte sad da se predomislite i da smanjite uticaj ženskog sveta na naslovne strane Računara Samo nemojte ni da računari pot puno nestanu sa vaših naslovnih strana

Goran Aćimović
Beograd

Ne boj se, računari neće nestati! Mi imamo dužnost da brinemo za sve vrste računara, pa je tako, svega puta, na naslovnoj strani, pored zanimljive devojke koju pominješ prikazan i specijalan model damskog malokalibarskog računara Sve na svome mestu

Hamlet igra Elitu

Staus konzervativne avangarde dokazali ste poslednjom naslovnom stranom zakopčanom do levog uha (košulja) mada dolazim do zaključka da vam ekipa u štampariji previše čita Bukovskog i suradnika Onako nemoralan prikaz ženske podlaktice i to sa ruskim satom i paralelom u obliku značka komesara Police Departmenta ostavilo me sa trgovinskim deficitom od 300 dinara

Kad biste malčice skratili one sažetke igara a ne da meni ispali 6 sijalica od 150 kW dok dođem do pola Dobro ime tu umjetničkih elemenata ali bateli Nije važno

Divno je kad naša hakerska opozicija pokaže humanu solidarnost sa Francuskom i opredeli se za orica ili u žargonu o bike Jugo bikovi uspaljeni erotičnošću crvenih tipki nisu odoleli pa smo sada nakon Mongolije i mi postali kompatibilni sa Ćilavom Gvinejom Bisao Dobro sada i nije tako loš očajan je

Onaj ekranski editor vam je bomba Ode mi dioptrija sa −1 na −9 i sad više ni kontektne ne mogu nosit Još i konjunktivitis! Sad se vjerojatno pitate gde se gasim Odoh je društvo ali vi ste još uvijek the best

Goran Štimac
32 divizije 19
41000 Zagreb

Bili smo iskreno uvereni da će kad-tad početi da nam stižu i paničke pisma i, gle čuda, to se i dogodilo. Ovo je jedno od takvih poetskih pisama koja ćemo kroz koju godinu objaviti u zbirci pod nazivom „Umetnost rasstavanja" Rasel

Antibiotik za zvrkove

Javljam se povodom pisma Tomislava Vazdare Ja sam također kupio zvrkove Srećom posjedovao sam Large Cargo Bay Ubrzo sam primjetio što se događa i napuslio sam spremište do kraja čvrsto riješen da se zaletim u neku planetu i završim igru Odlučio sam da kamikaza akciju sprovedem na maloj planeti koju do tada nisam bio vidjeo(?!) Krenuo sam hiperbrzinom i zaustavio se vrlo blizu Od muke sam pogledao da li su zvrkovi već izašli Nemalo sam se iznenadio kad sam vidjeo da su nestali Prostor je bio slobodan za robu

Radoslav Dejanović
V Kovačića 5
Zagreb

Čovek se uči dok je živ. Ili bar dok je komander

Randavu sa Jelenom

Čitam vaš i naš list od prvog broja do poslednjeg koji je trenutno ispred mene na stolu Veoma mi se sviđaju rubrike Load Peek&poke show i svi tekstovi MNOGOPOŠTOVANE Jelene Rupnik i malo manje poštovanih ostalih

članova redakcije (bez uvrede naravno). Ne bih da šaljem Jeleni nikakve poljupce preko redakcije već bih samo molio redakciju za informaciju o krugovima u kojima se J R kreće kako bih eventualno mogao sa njom da razmenim mišljenje

Voleo bih da saznam na koji način se dolazi do potrebne valute za nabavljanje naših mezimaca jer koliko je meni poznato devizno tržište je otvoreno samo za radne organizacije istina čuo sam za neke laserske štampače koji vrlo verno preslikavaju izvesno zeleno naslikane i ostale papirčiće (Nagradno pitanje Postoji li interfejs pomoću kojeg pomenuti štampač mogu povezati sa spektrumom ?)

Hteo bih nešto da poručim drugu Suadu B. koji tako slobodno izjavljuje da mrzi spektrum i njegove vlasnike Hteo bih da konstatujem žalosnu činjenicu da poznajem par novopečenih komodoriste koji su se bo juče ponosili svojim dugama a sada ih sramežljivo teraju u zapećak uz prezrivu tvrdnju da se gumice koriste samo za tegle sa slatkim E pa ja svim tim zlobnicima poručujem da se gumice mogu upotrebiti i u preventivne svrhe od prevelike navale raspusnih komodoriste

Za kraj bih pozdravio sve članova uredništva i redakcije uz molbu da u nekom broju objava onaj (barem meni) zagonetni način na koji se mogu u program ubaciti pokovi šifre i ostala preko potrebna pomagala za svršetak igara Jedino bih još mogao da zamolim za preporuku za brojeve sledećeg kola Lotoa — možda bih lako mogao da poboljšam svoje finansijske i kompjuterske pozicije

Vanja Đurđević
Proleterskih brigada 6/9
11001 Beograd

P S Malo je trebalo da zaboravim glavno pitanje Kako se postaje saradnik Računara ? Naravno honorarni Šta ćete svi smo mi materijalisti — velika je inflacija!

Da nismo, pre jedne godinu dana, nabavili jedan od tih laserskih štampača koji tako dobro preslikavaju zelene novčanice, teško da bismo mogli da nabavljamo sve one lepe mašine koje prikazujemo Redakcija „Peek & poke answe' razmišlja o raspisivanju nagradnog konkursa na temu „Za što sve mogu da služe gumice? Da li bi bilo odziva? Saradnik „Računara se postaje veoma lako. Prvo nam pošalješ nešto što si ti napisao ili isprogramirao. Onda mi pogledamo da li nam to odgovara Ako nam odgovara, mi to najčešće objavimo Ako objavimo, onda i platimo i tako dalje

Manje-više oštra kritika

Pored pohvala vreme je da najzad dobijete i poneku manje-više oštru kritiku Ne bojte se nije toliko strašno! Drug Željko iz Banjaluke je sasvim u pravu! Uradite nešto sa tim ivicama! Devedeset i pet odsto računardžija najviše voli da se igra sa svojim mikro ljubimcima Zato ne dajte da izađe broj bez bar 5—6 podugačkih opisa najnovijih igara Izaberite ponovo 50 najboljih za spektrume i komodora dodajte još par onako kvalitetnih strane sa lepim sličicama kao u poslednjem broju Taman me za nešto zainteresujete nekim kratkim opisom malog novog hard-dodatka najavite opširniji prikaz u sledećem broju i — ništa! Ja čekam!

Aleksandar Pantić
Dušana Popovića 3
11050 Beograd

Drago nam je kada nas kritikuju ljudi iz čijih pisama se vidi da nas i vole Nije bilo strašno. Naprotiv Neki delovi „Računara" će da se razvijaju upravo onako kako bi se tebi svidelo Naročito kad je reč o ivicama Pričekaj

Humanizam i renesansa

Samo da znate da me je vaša trinaestica (i naša kad rikneš tri konja) skoro stajala glave Naime umalo sam skočio kroz prozor od radosti kada sam pročitao da ćete objavljivati listinge ekranskog editora Vlade Kostića

Šokirao sam se kada sam pročitao bogovor pismu Suada Bajtovića Kako možete da hvalite to? Ne shvatam kako se može mrzeti jedan mikroprocesor? Svaki procesor a samim tim i računar vredi onoliko koliko vredi njegov programer ili korisnik Prema tome 6502 druga Bajtovića je bezvredan koliko i njegov komodor Slažem se sa tim da fali Biblioteka u dvanaestici samo mi nije jasno zašto se on uzrujava zbog toga (tamo neće naći mapu ni jedne igre) Zato predlažem drugu Bajtoviću da nastavi uništavati invajdere i da se ne meša u stvari za koje nije dorastao Nadam se da će ljudi napokon shvatiti da računari (ne mislim na list) ne predstavljaju sredstvo za iskazivanje neprijateljstva prema okolini

Zoltan Pastor
Laze Mančića 1/8
Novi Sad

Jako nam se svideo humanistički završetak tvog pisma, ali da li je prethodni deo pisma u skladu sa njim Hajde priznaj, ipak nisi tako mislio „Računari" uče da kad ti neko podmetne bajt, ti treba da mu pružiš još jedan registar

Suad i Suada

Opet onaj što ga Kelly pozdravlja Tako je Suad Bajtović iz Olimpijskog grada (ko je pomenuo Beograd?).

Momci iz Yugoslav Cracking Servicea su mi poslali pismo u kojem piše 128 is the bestlll (u prevodu — 128 je najbolji!) To pismo ću da pokažem svima u RAK-u AH da — RAK To je jedini Računarski amaterski klub u BiH Imamo QL, četiri spektruma i jednog C-64 sa diskom Osim toga, imamo oko 600 članova Od njih je 589 muških i onda neki tamo kažu da je zabluda da kompjuteri nisu za žene Spektrumovci iz RAK-a su hteli na krst da me prikucaju zbog nekih tamo izjava u kojima ih napadam Tužna ironija je to da je 99% pametnih ljudi u klubu opredeljeno za spektrum Ja ipak volim pametne ljude i ispade da ne mrzim sve spektrumovce

Neki kažu da je 13 nesretan broj ali je vaš trinaesti broj ispunio sva očekivanja Ja stvarno nemam ideju zašto se kupuju ostali računarski časopisi pored vas živih Razbarušeni sprajtovi su super (spektrum nema pokove) Ponavljam vi ste najbolji računarski časopis u Kumovoj Slami — posle PCW-a

Još nešto gde nestade Jelena Rupnik u 13 broju? Na nju liči ono Računari i žene ali to nije potpisano Osim toga to piše udata žena a sumnjam da je Jelena toliko netar da se uda pored tolikih živih čitalaca Računara Uz put namojte da objavite njenu sliku Neka svaki ubogi pirat i haker (spadam u obje kategorije) zamišlja svoju verziju Jelene Rupnik! Ne može ona da bude po volji svim hakerima i piratima i šta ste onda postigli (jesam dosadan) Živela Jelena Rupnik boginja zaštitnica svih hakera i pirata

Smatram da ovo pismo nije tako dobro kao prošlo ali sjetite se onog oprobanog TV pravila u pilot emisiji stavili sve najbolje da emisija krene a onda kako bude.

Suad Bajtović
Palmira Toljatija 136/V
71000 Sarajevo

P S Imam dva iz rukopisa A uopšte se ne primjeti Ne vole me nastavnici

Vlada kao turista

Primetio sam da je Svet kompjutera nedavno objavio jedan kraći tekst vašeg para Vlade Kostića poznatijeg kao THE Haker Pitam se da li to znači da je Vlada promenio boje kluba i počeo da piše i za druge ili je to možda neki drugi Vlada Kostić ili ja nisam dobro video potpis Vi verovatno znate šta je od toga istina pa vas molim da mi objasnite To nije neko značajno pitanje ali me puno interesuje

Saša Žeković
Zagreb

Stvar je dosta jednostavna. Vlada je naš ekskluzivac i nije promenio boje Tekst koji ste videli objavljen je s našim dopuštanjem

Sedam gnevnih čitalaca

Upravo sam završio sa prvim pregledanjem četrnaestice i sa žaljenjem sam ustanovio da u njoj nema mog pisma To me ipak nije pokolebalo i pišem ponovo Možda će moje pismo delovati isuviše kritizerski ali to je zbog toga jer sam primetio da se mnogo više osvrćete na kritikovanja nego na molbe

— Kao prvu primedbu naveo bih moje mišljenje u vezi sa poskupljenjem lista Računari sada koštaju lepih okruglih tri stotine dinara imaju raznovrsne teme koje su manje i više zanimljive (svaka vam čast na izboru) i sve to je smešteno na šezdeset i osam strane (sa koricama). Galaksija ništa manje dobar list sa obiljem zanimljivih tema — nema nezanimljivih — na istom obimu stranica i sa mnogo ozbiljnijim štampanjem posebno fotografije košta samo sto pedeset novih dinara — znači upola jeftinija

Sve zaključke prepuštam čitalačkom auditorijumu pošto vi obično nemate komentara za ovakve primedbe Napominjem još jednom da Računare čitam od prvog broja bez obzira na poskupljenja koje smo sa vaše strane bili izlagani

— Kao drugu primedbu bih izneo vaše lično razmetanje u nekim prilikama Ja lično nemam ništa protiv bilo kog računara (osim par izuzetaka) ali mi jako smetaju omalovažavanja i hvaljenja bo strane komodorista na račun nas SPEKTRUMAŠA da ostale i ne spominjem Što se vas lično tiče rekao bih samo toliko da mi nije jasno kako ste mogli da se hvalite da cela redakcija poseduje BBC računare (koliko li su vam samo platili!) Verovatno svaki član redakcije ima par po dve tri tetke pet šest ujaka na privremenom radu po Engleskoj i Nemačkoj zar nali!

Treća stvar bi mogla da bude u vezi s načinom nabavke opreme Na časti vam služi članak u najnovijem broju! Uostalom kada već toliko piratujemo programe što ne bismo švercali i računare (makar delovima) Sreća vaša a i naša što carinici čitaju samo inostrane cenovnike Baš razmišljam kako sad da uvezem jedno deset pomenutih IBM klonova

— Mislim da bi bilo vrlo zgodno da u nekom od brojeva pomenete kako se na neki instant način može doći do para za svu divotu iz rubrike Šta ima novo — Siguran sam da bi tu jako pomogla iskustva svih članova redakcije i njenih saradnika posebno po pitanju deviza jer ja nemam nikoga u inostranstvu (tetke babe ujne stričeve)

Čini mi se da SIV i narodna banka još ne vrše prodaju deviza na malo (kad počnu obavestite me).

— Ima još par nepoznatih za koje smatram da mogu biti interesantne za sve nas pa čak i veoma korisne za same Računare

Predlažem vam da osnujate sekciju za vaše ljude iz štamparske službe gde bi oni išli na praksu štampariju koja štampa jedan ugledni i dobro poznati list za računare u bratskoj nam republici Sloveniji Da bi odziv vaših radnika bio veći može se organizovati izlet do obližnjeg nam Trsta besplatno naravno (sve u duhu stabilizacije) Pitam se na vas samog sebe kako Slovenci mogu samo da čine sve takve stvari a vi ne možete ccc

Na kraju bih vas samo zamolio da mi oprostite na neuredno otkucanom tekstu jer nisam bio u mogućnosti da ovaj tekst obradim preko WORDSTARA kako to radi drug Srđan Kosovac (neka bude IBM) čak je i ova mašina pozajmljena Šta mi je?

Drugarski vas pozdravljam s nadom da ćete kvalitet štampe poboljšati i pre 2000 -te godine

Vanja Đurđević

Čitaoci „Galaksije dobijaju svoj list na kioscima jeftinije nego što on košta u proizvodnji. To je rezultat svesnog ulaganja OOUR „Duga" u popularizaciju nauke. „Računari" izlaze na komercijalnoj osnovi — oni moraju da žive ili od onih zbog kojih i postoje, dakle od svojih čitalaca ili od — oglasa Da li bi bio srećniji da je vreme stalo sa računarima ZX 81 i „galaksijom"? Grafičari BIGZ-a cene umešnost svojih slovenačkih kolega Danas! nam ipak, koji tabak onog finog papira na kome rade Slovenci pa ćeš videti da ni mi nismo ništa lošiji

štampači

Ovaj prikaz je uglavnom, usmeran na štampače koji su danas interesantni kako za programere i hakere teko i za one koji planiraju poslovnu primenu kompjutera matrični štampači koji ispisuju 80 normalnih znakova u redu poseduju NLQ opciju koštaju 200—500 funti i nude danas praktično neophodnu epson kompatibilnost Ukoliko za to buda interesovanja i mogućnosti pripremićemo i prikaze ostalih tipova printera (štempači se lepezom električna pisaće mašine sa interfejsima ink-jet printeri itd) za neki od sledećih brojeva našeg časopisa

Pre nego što pogledate našu veliku tabelu napomene o pojedinim modelima i njihove otiska posvetićemo određenu pažnju karakteristikama štampača i pokušati da objasnimo termine koji se često sreću a retko razumeju Svi štampači koja prikazujemo su **matrični,** što znači da se otisak na papiru formira udaranjem iglica u specijalnu traku Obzirom da se znaci sastoje od tačkica njihova čitljivost uvek teži da buda neprijatno slaba pogotovu za ljude koji nisu navikli na ovakav način pisanja Struktura će se jasno manje primećivati ako su tačke gušće raspoređene što znači da štampači koji imaju više iglica nude kvalitetniji takst U našoj tabeli smo, međutim odustali od navođenja veličine takozvane „matrice karaktera jer je ona jednaka za sve modele koje pominjemo 11*9 Razloge za ovakvu unificiranost nije preteško pogoditi svi su modeli epson kompatibilni što znači da programi koji definišu karaktere na epsonu moraju da rade i na njima Davno je uz to procenjeno da je matrica 11*9 dovoljna za normalne' primene za specijalne primene su rezervisane i specijalne opcije

Strogo kontrolisani štampači

Početni redovi tabela, osim imena štampača (1) navode i njegovu englesku cenu (2) koja uključuje i VAT od 15% Zatim prelazimo na tehničke podatke vrata 3 nabraja brzinu štampača koju proizvođač navodi u specifikacijama. Ovaj je podatak potpuno beskoristan i verovatno nikome na svetu nije jasno kako se do njega dolazi realne brzine su daleko manje Zato smo usvojili nekoliko dodatnih testova. **Svi** štampači o kojima govorimo omogućavaju automatsko testiranje dovoljno je da ubacite papir i traku i, uključujući štampač, pritisnete neki specijalni taster Printer će tada, sve dok ga ne isključite u sukcesivnim redovima ispisivati svoj set karaktera i tako vam omogućiti da kontrolišete ispravnost uređaja Obzirom da je ovo štampanje izuzetno brzo posvetili smo mu vrstu 4 primetićete da je čak i takva brzina daleko manja od one koju navodi proizvođač Rezultati se dalje smanjuju kada dovedemo štampač u realne uslove vrsta 5 daje brzinu rada na specijalno pripremljenom sintetičkom tekstu koji je dug 5000 znakova (oko 2 5 šlajfni) Tekst je *sintetički* utoliko što sadrži minimalan broj blanko simbola što su redovi podjednakih dužina i što sadrži minimalan broj blanko simbola što su redovi podjednakih dužina i što se piše samo jednom vrstom slova Vrsta 6 daje rezultat koji će vam u praksi najviše značiti brzina štampanja standardnog teksta se novim redovima podvlačenjima, prelascima na kurzivna *(italic)* slova, korišćenje istaknutih naslova i podnaslova Tačnost merenja je diktirana tačnošću štoperice pošto ne bi bilo fer koristiti časovnik ugrađen u računar svaki štampač ima bafer određena veličina (21) tako da je računar slobodan mnogo pre nego što štampač završi posao

Iako se još nismo detaljno bavili NLQ modom verovatno znate da moderni matrični štampači nude daleko superiorniji otisak dobijen na račun brzine rada Gubici u brzini rada su, kao što se vidi iz vrste 7 ogromni standardan tekst od 10000 znakova se u standardnom *(draft)* modu ispisuje brzinom od stotinak karaktera u sekundu dok isti tekst u NLQ modu zahteva sekundu za svega petnaestak znakova!

Svaki se štampač jasno nalazi pod kontrolom računara umesto slova kompjuter može da pošalje propisanu sekvencu kontrolnih kodova i tako izazove promenu tipova slova prelazak na novi list i slične stvari Kontrolisati štampač posredstvom računara nije međutim baš uvek komforno često ćemo poželeti da preskočimo nekoliko redova ili pređemo na sledeći list a da pri tom ne pamtimo komplikovane Escape sekvence i ne kucamo duge komandne linije Zbog toga je na kutije svih modernih štampača ugrađeno po nekoliko tastera Uz neizbežni *On-Off* štampači koji slede Epsonove standarde imaju dirku *On Line* pomoću koje se štampaču zabranjuje da prima dalje znakove od računara (zbog čega je to korisno? Ponekad će vam zatrebati da promenite list papira ili obavite telefonski razgovor u toku koga vas buka neće ometati) taster *Line Feed* pomoću koga štampač preskače jednu liniju i najzad, taster *Form Feed* koji izaziva prelazak na prvi red sledeće stranice Ni jedan od štampača koji pominjemo nema mogućnost da detektuje perforaciju između stranica što je odlika nekih skupih profesionalnih modela kraj strane se prepoznaje tako što je štampaču poznata njena dužina i time što pretpostavlja da se po svakom uključivanju nalazi na početku novog tabaka Ukoliko vam se dakle dogodi da isključite štampač koji nije završio sa ispisivanjem stranice, moraćete da okrećete papir pomoću uvek prisutne ručice i da uključite printer tek kada perforacija prođe ispod glava

Osim standardnih kontrola koje se podrazumevaju pa ih nije imalo smisla nabrajati u tabeli neki štampači omogućavaju da se kontrolnog panela biraju NLQ slova (8) pa čak i da se izabere mod i tip slova (9)

Modovi, tipovi i širine

Domaći ljubitelji računara (pa čak i kompjuterski profesionalci) često mešaju termine „mod' i ‚tip slova' *Tip* (ili u stranoj literaturi, *font)* označava oblik slova, na primer uspravna *(plain)*, kurzivna *(italic* — nikako nismo uspeli da saznamo da li su termini *kurziv* i *italic* sinonimi ali u svakom slučaju jako liče) ili gotica. Reč *mod* označava način ispisivanja slova i daje predstavu o kvalitetu rezultujućeg teksta uobičajeni su termini *normalni (draft) NLQ (Near Letter Quality)* i *LQ (Letter Quality) mod* Tako možemo imati uspravna slova pisana u draft modu ili kurzivna slova u NLQ modu

Da bi stvar bila još komplikovanija, uvodimo i termin *širina slova (pitch)* Standardne širine slova su *Pica* (10 znakova na inč širina papira), *Elite* (12 znakova po inču pri čemu kod nekih štampača *Elite* slova imaju i drugačija oblika što bi značilo da se radi o novom tipu) *Condensed* (17 karaktera po inču) *Enlarged* (5 znakova po inču) i *Condensed Enlarged* (8 5 znakova po inču) Neki štampači uz standardni indikator koji pokazuje da li su ‚On Line , imaju i posebne LED diode koje prikazuju da li im je nestalo papira (10) u kom se modu nalaze (11) pa čak i zvučni alarm (12) koji se oglašava kada nešto nije u redu na primer kada se papir umrslo

Uz kontrole na kućištu svi štampači poseduju i grupe mikroprekidače kojima možete da izaberete njihovo početno stanje Ukoliko na primer uglavnom radite se kondenzovanim slovima kako biste u svaki red mogli da smestite 132 znaka jednom ćete odvrnuti nekoliko zavrtanja, skinuti kućišta štampača i pomeriti mikroprekidač koji se odnosi na kondenzovani mod Tako ćete izbeći potrebu da po svakom uključivanju štampača šaljete sekvencu kontrolnih kodova koji ga prebacuju na kondezovana slova ali ćete ako vam jednom zatrebaju normalna slova morati da šaljete drugu sličnu sekvencu Opšta karakteristika mikroprekidača je, dakle da se stanja koje je njima postavljeno može poništiti ili promeniti softverski Za jugoslovenska potencijalne vlasnike štampaće su posebno važne kolone 13 i 14 naše tabela kod većine printera se mikroprekidačima može izabrati dužina stranice (naš list ima 72 reda a engleski i američki 66) i rastojanja između linija **Svi** pobrojani štampači imaju mikroprekidač kojim se može zahtevati preskakanje perforacija na kraju svake stranice

Vrste papira

Laicima je najnormalnije da štampači koriste običan papir A4 formata Ovaj se papir kad i na pisaćoj mašini kreće oko

Ako se izuzmu prikazi ponekog popularnog modela, „Računari" se još od prvog broja nisu ozbiljnije bavili štampačima. Vremena kada su nastajali „Računari u vašoj kući" su sada daleko iza nas, pa nam čitanje tadašnjeg napisa izmamljuje podosta osmeha: o printeru se više ne sme govoriti kao o nedostižnoj spravi potrebnoj jedino onima koji primenjuju kompjutere u radnjama male privrede — oni danas predstavljaju nezaobilazan inventar svakog kućnog računskog sistema. Štampači izvanrednih karakteristika se na stranom tržištu nude pod izuzetno povoljnim uslovima. U ovim i sledećim „Računarima" dajemo detaljan prikaz najkvalitetnijih i najprivlačnijih modela, a pripremamo i nekoliko ekskluzivnih priloga za one koji ih već imaju — detaljno uputstvo za upotrebu štampača iz „epson" standarda i uputstvo za ubacivanje YU slova u njihove ROM-ove.



pokratnog valjka koji ja ugrađen u standardnu opramu većina štampača iz arednja klase (16) Nije, međutim, baš prijatno koristiti A4 listove štampaču je, da bi ispisao jedan takav list potreban otprilika jedan minut što znači da ćete svakoga minuta morati da ubacujete novi list! Posla izvesnog vremena ćete utvrditi da ubacivanje listova odnosi više vremene nego samo štampanje, pa ćete potražiti racionalnija rešenje Jugoslovenski vlasnici računara obično koriste perforirani kompjuterski papir (u atranoj literaturi *fon-fold paper)* koji se obično isporučuja u sanducima od kojih svaki sadrži po 2000 međusobno povezanih listova papira proširenih perforacijom sa svake strane (sanduk od 2000 listova košta oko 5000 dinara i traja prilično dugo, autor ovoga teksta nikada nija naročito štedeo papir a ipak mu svaki sanduk potraja po desetak meseci) Štampač koji treba da prima ovakav papir mora da bude opremljan takozvanim *traktorom* koji je obično (15) uključen u standardnu opremu Širina listova koja prima traktor se može menjati pomeranjem vodica za papir ali ja ovo pomeranja kod nekih starijih modela prilično loše rešeno tako da štampač može prihvatati samo papir u uskom rasponu širina Moderniji printeri (16) mogu da primaju papir svih širina do maksimalne (18) tako da se mogu ispisivati čak i uske nalapnice za koverte što je od značaja za određene poslovne primene

Najjeftinija rešenje ja korišćenje rolne papira, za šta je, kod nekih printera, potrebno dokupiti poseban dodatak (17) Ovaj je papir prilično nekvalitetan i ima ga smisla koristiti samo ako se rezultati koriste za lična potrebe vlasnika računara Jasno je da pre početka rada sa rolnama treba onemogućiti preskakanja perforacija jer perforacija i ne postoji!

Cepanja papira je veoma bolna tačka za većinu štampača u normalnom radu često je potrebno odcepiti upravo ispisani list papira bez potrebe da se izbacuje još jedan prazan i, po mogućnosti, bez potrebe da korisnik ustaje sa stolice Pitanja cepanja papira je razrešeno na najrazličitija načine (20) i ma kako banalno izgledalo predstavlja jedan od veoma ozbiljnih parametara koji treba uzeti u obzir pri izboru štampača.

Tehnologija štampanja

Posvetimo malo pažnje stvarima koja se događaju od momanta kada računar pošalje jedan znak štampaču od momanta kada se taj znak pojavi na papiru Veze računara i štampača se obično zasnivaju na principu takozvanog hand shaking-a osim linija za prenos podataka računar i printer su povezani linijom koju zovemo *ready-busy* Preko ove linija štampač signalizira računaru da je spreman da primi sledeći znak koji mu zatim biva i poslat Taj znak putuje pravo u takozvani *bafer* specijalni RAM koji je ugrađen u štampač Kada se bafer popuni (njegova veličina (21) varira od modela do modela), računar mora da čeka da se neki znak odštampa da bi u baferu bilo mesta za sledeći Veliki bafer omogućuje da se računar , oslobodi za druge poslove mnogo pre nego što se štampanje završi ukoliko uglavnom pišete kraća pisma i izveštaje, isplatiće vam se da investirate u veći bafer, jer će tako štampač moći da radi dok vi (ako vam ne smeta buka) pišete sledeće pismo Na baferu se sa druge strane može dosta i uštedeti računari sa pristojnim RAM-om koji su opremljeni fleksibilnim operativnim sistemom mogu da organizuju komunikaciju sa štampačem preko prekida (interapta) i tako paralelno obavljaju dva posla koristeći višak sopstvenog RAM-a kao bafer za štampač Ovakva metodologija je kod Acornovih računara koji su dopunjeni „bočnim" RAM-om dovedena do savršenstva

Pre nego što se oduševite veličinom bafera nekog od modela koji opisujemo razmislite o jednom ograničenju uobičajeno je da se bafer za štampač koristi i za definicija karaktera tako da se pomeranjem specijalnog mikroprekidače bira jedna od ova dva namena Kako će vam s vremena na vreme zatrebati naša latinična slova i kako će vas mrzeti da par puta nedeljno

UPOREDNE KARAKTERISTIKE ŠTAM[illegible]ACA

		Epson LX 8	Epson F	Mannesmann T 11 5	K	P
	[illegible]lijentaciona cena (funti)	350	526	4 2	300	338
	BRZINA					
	[illegible] pecifikaciji PS	100		1 0		
	Brzina self testa	77	1			
	[illegible] ticki tek t		1			
	[illegible] lni ek t	57	79			
	NLQ reel i tekst	1				
	KONTROLNI PANEL					
	[illegible] NLQ mo		da	da		
	[illegible] tipa lova	1				
	INDIKATORI					
	[illegible] papira	da		3		
	[illegible] NLQ mod	n				
	Zvučni alarm					
	MIKROPREKIDAČI					
	Izbor dužine [illegible]		ne			
	Izbor [illegible]					
	[illegible] redova	da	ne			
	PAPIR					
	Perforirani papir					
	A4 listovi				da	
	[illegible]	da	da			
1	[illegible] širina (mm)	2[illegible]	[illegible]	2		
	Širina traktora					
	se i [illegible]	da	ne	ja	da	d
	[illegible]panje li tov					
	KVALITET TEKSTA I KOMFOR					
	Bu[illegible]	1			5	
	Br j pecijalnih					
	[illegible] trani		16	9		
23	[illegible] slova	super	[illegible]	[illegible]per	do[illegible]	
24	D[illegible]der	[illegible]ori	[illegible]			
	[illegible]					
2	Tačkasta struktura (Draft)	ok	ok			
2	Tačkasta struktura (NLQ)	dobra	Ok	[illegible]upo	dobra	
2	Broj karbon kopija	1	1			
	ODRŽAVANJE					
29	Umetanje listova papira	lako	Ok	lako		
30	Umetanje perf papira		teško	ok	[illegible]	
31	Promena trake	vrlo lako	[illegible]	[illegible]	la[illegible]	
32	Da li su ruke prljave?	ne	veoma	ne	ne	
33	Uklanjanje zgužvanog lista	teško	teš[illegible]			
	UPUTSTVO ZA UPOTREBU					
34	Detalji	super	super	[illegible]	da	mnogo
35	Sadržaj	da	da	da	[illegible]	
36	Indeks	super	super	[illegible]		
	INTERFEJSI					
37	Serijski	opcija	[illegible]	[illegible]terna [illegible]	[illegible]	op[illegible]
38	IBM kompatibilan	ne			[illegible]	
	BUKA					
39	Koliko vrata?	2			2	

rasklapate štampač bafera ćete se najverovatnija zauvek odreći!

Pošto jedan znak iz bafare doda na red, printar će ga analizirati i konstatovati da li se radi o običnom znaku ili o kontrolnom kodu. Ukoliko se radi o znaku treba još razmisliti o načinu na koji će se on što brže ispisati. Svi štampači koje opisujemo imaju mogućnost štampanja u dva smera uz takozvano logičko traženje: glava će se pre nego što počne da ispisuje red, pomariti na onaj njegov kraj koji joj je bliži.

Nailazak na kontrolni karaktar signalizira štampaču da iz bafara uzme i sledeće karaktare koji će zaokružiti takozvanu *Escape sekvencu*. Escape sekvanca je niz kontrolnih kodova koji počinju sa „Escape" (27) i koji kontrolišu štampač ili mu naređuju da buduće znakova ispisuje na neki specijalan način. A ti specijalni načini su, uz one koje smo već pomanuli *Proportional Emphasized* i *Double strike*. Proporcionalno razmicanja je lako razumeti: širina slova su različite tako da je slovo I dalako uže od slova 'm'. Iako proporcionalno razmicanja znatno doprinosi dopadljivosti teksta, vlasnici računara ga veoma retko koriste pošto ga ne podržava praktično ni jedan tekst procesor. Svaki tekst procesor, naime, omogućava korisniku da uravna desnu ivicu teksta pri čemu jsano, pretpostavlja da su sva slova jednako široka. Nemoguće je, dakle, urevnavati desnu ivicu i koristiti proporcionalno razmicanje. Ukoliko isključite uravnavanja, neki će redovi izgledati prekratko pa će tekst biti pomalo smešan. Sve u svemu, proporcionalno razmicanje slova nije karakteristika štampače kojoj traba obraćati mnogo pažnje!

Izgleda da malo ko zna u čemu je tačno razlika između Double Strike, Emphasized i NLQ štampanja. Autor ovoga teksta mora da prizna da je i on od skora spadao u tu apsolutnu većinu i da se nije ni najmanje zabrinjavao zbog svoga neznanja. Nedavno smo kroz uputstvo za printere firme *Micro Peripherals* imali priliku da upoznamo sličnosti i razlike između ova tri načina štampanja: sličnost je to svi znaju u tome što glava u sva tri slučaja dva puta prelazi preko svakog reda. Kod *Emphasized* (ili kako ga ponekad nazivaju, *Bold*) štampanja svaka se tačka ponavlja uz malo horizontalne pomarenje tako da je horiznotalna linija slova T vrlo gusta (ne primećuje se njene tačkasta struktura), dok je vertikalna linija istog slova nešto debija ali i dalje tačkasta. Kod *Double Strike* štampanja između dva prolaza dolazi do minimalnog vertikalnog pomeranja tako da je vertikalna crta slova T gusta, a horizontalna debela ali tačkasta. NLQ štampanja, najzad, podrazumeva veoma precizno pozicioniranje glave tako da i vertikalne i horizontalna linija daju utisak celovitosti. U NLQ modu se, osim toga koriste posebni oblici slova definisani (obično) na metrici 23*16: slova su pripremana tako da imitireju električne pisaće mašina i štampače sa lepezom. Svi štampači pomenuti u tabeli omogućavaju proporcionalno razmicanje, 'italic', 'emphasized', 'double strike' i NLQ' štampanje kao i pisanje indekas i eksponenata mada ne obavezno i u NLQ modu: videćemo na primer, da uz Canon PQ1080A morate da kupite dodatni ROM da biste štampali kurzivna slova u NLQ modu. Svi štampači osim toga omogućavaju korisniku da definiše specijalne znake (na primer, naše



latinična slova), kako u običnom tako i u NLQ modu Za definisanje NLQ karaktera je, međutim, potrebno dosta RAM-a pa neke štampače treba dopuniti 6264 CMOS čipovima koji se utiču u predviđene podnožja.

Umesto da se mučite sa definisanjem slova možete da izvadite EPROM-e u koje su upisani oblici znakova (u štampač su obično ugrađena tri do četiri EPROM-a 2764 ili dva EPROM-a 27128, Epsonovi štampači koriste ROM-ove koje ne možete da brišete, ali možete da zamenite EPROM--ima) i da, umesto neke od specijalnih azbuka (22, obično se zamenjuju švedska slova) uvedete naša slova Iako je za ovakvu hirurgiju potrebno određeno strpljenje ona nije teška, pa ćemo joj posvetiti malo prostora u nekom od sledećih brojeva „Računara"

Kvalitet otiska

Kvalitet otiska je verovatno, najvažnija odlika jednog štampača ako kupite štampač sledeći logiku „nije važno kako dokument izgleda samo da štampač nije skup", očekujte u bližoj budućnosti nove troškove za nov printer Vrsta 23 govori o obliku slova, vrsta 24 o slovima koja se protežu ispod osnovne linije kao što su j g p q i slična (ova se slova u stranoj literaturi označavaju kao *descenders*) dok vrsta 25 opisuje kvalitet podvlačenja Jasno je da je svaka od ocena u ove tri vrste plod subjektivnog utiska Vrste 26 i 27 u tabeli najzad, govore o tome koliko se tačkasta struktura slova primećuje u normalnom odnosno u NLQ modu

Mnogi proizvođači tvrde da njihovi modeli mogu da proizvedu određen broj „karbon" kopija Ta se kopija međutim ne prave pomoću „običnog" indiga — treba kupiti poseban papir sa indigom ili papir kod koga su listovi pravljeni tako da se jedan automatski otiskuje na sledećem Zato smo uveli vrstu 28 koja prikazuje broj običnih karbon kopija koje štampači prihvataju Taj broj, kao što se vidi retko prelazi jedan

Upotreba i uputstvo

Za razliku od računara štampači zahtevaju određeno održavanje treba menjati papir, montirati traku pomerati mikroprekidače odstranjivati „konfete" koje preostaju u samom štampaču i raditi mnogo sličnih stvari Vrsta 29 opisuje umetanje listova a vrsta 30 perforiranog papira, dok vrste 31 i 32 opisuju promenu trake Vrsta 33, najzad, opisuje situacije koje se neće dešavati previše retko gužvanje papira. Ma koliko štampač bio dobro izrađen i ma koliko pažljivo sa njim rukovali događaće vam se da se papir zaglavi Kod nekih je modela ova operacija sasvim jednostavna, dok kod drugih podrazumeva rasklapanje dobrog dela uređaja

Većina štampača ima **veoma** slaba uputstva za upotrebu kontrolni kodovi su samo pobrojani bez detaljnih opisa (34), nema indeksa (36) ni dobrog sadržaja (35), primere treba tražiti svećom, štamparske greške su brojne engleski toliko slab da i mi to primećujemo (knjiga su pisali Japanci) Teško je naći i drugu literaturu koja bi na popularan način upoznavala čitaoce sa funkcijama štampača pogotovu ako se radi o nekim „specijalnim efektima" kao što je definisanje normalnih i NLQ slova i crtanje slika. Obzirom da u Jugoslaviji postoji sve više „epson" kompatibilnih štampača, „Računari" planiraju da objave kompletno uputstvo za njihovu upotrebu prilagođeno kako početnicima tako i korisnicima kojima su potrebni dobri indeksi i tabela

Interfejsi i kompatibilnost

Svi štampači koje prikazujemo imaju ugrađen paralelni (Centronics) interfejs što znači da ih je lako povezati sa većinom danas popularnih računara koje vredi dopuniti štampačem (za ZX81 „spektrum" „elektron" i „galaksiju" treba dokupiti poseban interfejs što i nije neobično čudno je što se interfejs mora kupiti i za QL računar koji pretenduje na poslovna primene) Većina štampača (37) može da se dopuni serijskim (RS232) interfejsom koji je obično dopunjen dodatnim baferom Pitanje je, međutim koliko su takvi interfejsi potrebni Iako je uvreženo verovanje da serijski interfejs podrazumeva sporije štampanje nije tačno (, usko grlo" u paru računar—štampač nije komunikacija već brzina ispisivanja slova na papiru) nema nikakvog razloga da odustanete od paralelnog prenosa podataka, koji ne zahteva posebne troškove Prodavci kompjuterske opreme su uobičajili da uz štampač besplatno daju i kabl za povezivanje sa vašim računarom ali se na ovu konvenciju ne možete baš uvek osloniti za kabl ćete ponekad morati da doplatite 10—20 funti

Svi su štampači koje prikazujemo, kao što rekosmo na početku napisa „epson" kompatibilni ali neki (38) omogućavaju direktno povezivanje i sa IBM-ovim kompjuterima Ukoliko ste, dakle vlasnik IBM PC i želite da koristite FX80 moraćete da kupite hardverski ili napišete softverski konvertor koda

Mnogo buke . .

Štampači proizvedeni pre pet-šest godina imaju jednu užasnu osobinu prave toliku buku da vas nateruju da iskočite iz kože Uz ovakav štampač je pre svega, sasvim nemoguće raditi bilo šta osim nervirati se njegova je primena osim toga, iritirajuća kako za vaše ukućane tako i za (ni kriva ni dužna) komšije Obzirom da većina hakera ima običaj da radi noću nabavka štampača koji neće nikoga buditi i nije tako loša investicija Skoro bi se moglo reći da neki proizvođači namerno povećavaju buku koju njihovi modeli prave Štampači Star Gemini na primer aktiviraju svoju zujalicu kad god prelaze iz „Off Line" u „On Line" i tako standardnoj buci dodaju i stalno udaranje zvona

Buka se kao što znate meri decibelima ali je za takva merenja potrebno imati odgovarajuću opremu podatak koji bismo navali vam osim toga ne bi mnogo značio Zato smo izabrali jednu novu jedinicu za merenje buke koja bi se teško probila u SI sistem broj zatvorenih vrata Poslednja vrsta naše tablice, naime prikazuje broj vrata koja morate da postavite između sebe i štampača kako ne biste čuli njegov rad

O opštim karakteristikama štampača smo kao što ste i sami primetili već previše pisali pa ćemo preći na konkretnije stvari u sledećim „Računarima" prikazujemo Epsonove modele FX80 LX80 i LX85 štampače Kaga Taxan i Canon Mannesmann Tally 85 Smith Corona D200, Panasonic KX-P1092, Star SD10, Seikosha SP1000A, Samlaco DX66 Citizen 120D i Brother 1509

Dejan Ristanović

Naš test

igračke za odrasle

Ploter „Fišer"

Bilo je samo pitanje vremena kada će se neka firma setiti da kombinuje računarske primene i igre i tako stvori računarske igračke. Poslednjih godinu ili dve nekoliko proizvođača je izbacilo na tržište igračke koje se na ovaj ili onaj način dotiču računara. Firma „Fišer" je otišla i korak dalje. Počela je da proizvodi „Computing setove" — pakete delova za računarska pomagala. Najinteresantniji komplet sadrži delove za ploter-skener.

Centralni deo ploterskog kompleta je interfejs, koji i predstavlja osnovu svih Fišerovih „Computing setova". Uz interfejs se dobija i disketa na kojoj se nalaze programi potrebni za rad. Interfejs i softverska podrška se dosta razlikuju u kvalitetu. Dok je interfejs urađen dosta impresivno, programi za rad sa kompletom su prilično primitivni. Glavni mašinski program za upravljanje dat je u obliku DATA linija i programa svaki put mora ponovo da smešta podatke u memoriju, što usporava rad i prilično je nepraktično za korišćenje. Interfejsom se upravlja uz pomoć komandi SYS i USR — dakle direktnim pozivanjem mašinskih rutina. U svakom slučaju, ostaje vam ili da se snalazite sa onim što je dato, ili da sami napravite bolju programsku podršku, ako umete.

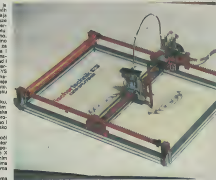
Sastavljanje plotera je, u najmanju ruku, — zanimljivo isto toliko koliko i rad sa njim. To i jeste ono što (pored relativno niske cene) predstavlja glavni kvalitet svih Fišerovih setova. Uz komplet se dobija detaljno postupno uputstvo koje se veoma lako prati.

Ceo uređaj se nalazi na pleksiglas ploči formata A3. U uglu ploče je elektromotor zadužen za pomeranje po Y-osi a na pokretnoj osovini motor za pomeranje po X osi. Naravno, reč je o veoma kvalitetnim step-motorima, i prenosnim mehanizmima u obliku puža pa je pokretanje veoma precizno.

Glava plotera (ili skenera) je veoma jednostavna. Najvažniji deo je elektromagnet koji se nalazi na samom vrhu glave. Njegova dužnost je da diže i spušta olovku plotera. To je urađeno veoma jednostavno: nosač olovke stoji na dve male opruge koje drže olovku odvojenu od papira. Kada ploter dobije komandu koja nalaže pisanje, uključuje se elektromagnet koji privlači ceo držač olovke nadole. Kada se magnet isključi, opruge ponovo vraćaju olovku u prvobitni položaj.

Veza između interfejsa i plotera je izvedena preko kabla od 24 žice. U uputstvu je posebno naglašeno da sve žice kabla moraju biti tačno određene dužine — ako to ne uradite po uputstvu imaćete velikih problema sa „oživljavanjem" plotera. On, jednostavno, neće hteti da radi.

Pametnom upotrebom možete postići da ovaj „skrpljeni" ploter daje rezultate koji, naizgled, nimalo ne zaostaju za nekim mnogo puta skupljim profesionalnim ploterom. U tome vam pomaže i program koji je dat zajedno sa uputstvom. Pomoću njega možete crtati pravougaonike, krugove i elipse, kao i ogroman izbor slova. Moguće je dobiti slova svih veličina, iskrivljena na sve moguće načine. Slova se mogu i rotirati i pisati u svim pravcima. Nabrojane mogućnosti otvaraju dosta prostora za zaista zanimljiv rad sa Fišerovim Frankenštajn ploterom, ali se njegovi potencijali otkrivaju tek u toku primene, kada pustite mašti na volju. Pri tom se ne plašite igranja. Ploter je upravo za to i namenjen. Svi delovi su zamenljivi, a izvestan broj rezervnih delova se dobija i u kompletu. Pojedini delovi su identični kao u nekim drugim setovima, tako da nije teško doći do njih kada ustreba.



Igračka ili korisno pomagalo: Ploter „Fišer" u akciji. Fotografija: Ljubodrag Simović

Drugu primenu ovog uređaja — skener — obradići nekom drugom prilikom. Transformacija plotera u skener nije komplikovana. Treba samo skinuti olovku i umesto nje staviti „čitač", promeniti softver i dovesti još dve žice do radne glave.

Fišerove „igračke za odrasle", kao što je ovaj ploter-skener, omogućuju mnogim ne preterano bogatim zaljubljenicima u računare da nabave i (po neki deo periferijske opreme za kojim odavno žude a nemaju ogromne sume novca da to i plate. Kvalitet opreme je sasvim na visini ako se izuzmu detalji koji direktno zavise od cene. Tako ovaj Fišerov ploter nikada neće izgledati kao „pravi", iako tako, možda radi. Jedna od stvari koja je mogla biti bolje rešena je i softverska podrška, ali bar tu imate mogućnosti da sami što-šta uradite, ako znate kako.

Bilo bi lepo kada bi i drugi proizvođači počeli da izbacuju sličnu opremu na tržište i da uskoro budemo suočeni sa mnoštvom sklopivih disk jedinica, štampača, svetlosnih olovaka i grafičkih tabilica. Stisnimo palčeve.

Vladimir Krstonošić

„128"
na sinklerov način

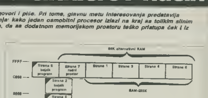
O „spektrumu 128" se sve češće govori i piše. Pri tome, glavnu metu interesovanja predstavlja organizacija memorije i večito pitanja: kako jedan osmobitni procesor izlazi na kraj sa tolikim silnim RAM-om? Odgovor je, kako izgleda, da se dodatnom memorijskom prostoru teško pristupa čak i iz mašinskog jezika.

Blokovi i stranice

Procesor Z80 normalno adresira lokacije između &000 i &FFFF što ukupno iznosi 65536 bajtova. Taj prostor se može podeliti na četiri bloka od po 16K pri čemu običan „spektrum" ili „spektrum +" najniži blok rezervišu za ROM a preostala tri za RAM. „Spektrum 128" tu ne može biti nikakav izuzetak, jer se bazira na istom procesoru. U bilo kom trenutku rada u sistem mogu biti uključena samo četiri bloka memorije od 16K.

Međutim, ukupan memorijski fond „spektruma 128" iznosi 160K, od čega 32K otpada na dva ROM-a, a 128K na osam blokova RAM-a od 16K. Memoriju tako možemo zamisliti kao knjigu sa desat stranica, pri čemu svaka strana sadrži 16K informacija. Procesor Z80 može istovremeno da čita samo četiri strane.

Prve dve strane (ROM) tretiraćemo posebno, a preostalih osam strana RAM-a numerisaćemo od 0 do 7.



Klip-klap

Jedan od ROM-ova je standardni bejzik ROM, dok drugi sadrži ekranski editor, rutine za rad sa zvukom itd. Po potrebi procesor uključuje jedan ili drugi, ali uvek se ROM nalazi u najnižem delu memorijske mape između adresa &0000 i &3FFF. Razume se, na isto mesto dolazi i ROM interfejsa jedan, ako je uključen u sistem. „Spektrum 128" prosto bira koji mu ROM kada treba i sa njim onda radi.

Međutim, kako stoji stvar sa RAM-om? Memorijska mapa nudi tri bloka od 16K počev od adrese 4000. Bilo bi logično očekivati da bilo koja od osam stranica RAM-a može doći u bilo koji od tri bloka mape. A nije tako. Blok između 4000 i 7FFF rezervisan je isključivo za RAM 5 a blok između 8000 i BFFF — samo za RAM 2. U vrh mape (C000 – FFFF) može se smestiti bilo koja stranica RAM-a (pa čak ponovo RAM 2 ili RAM 5)! Obično je, međutim, na tom mestu RAM 0, što sve odgovara standardnoj konfiguraciji „spektruma" i „spektruma +".

Noviteti nastaju kada korisnik pozove ekranski editor. Editoru je potreban dodatni prostor za pamćenje sadržaja ekrana, a u tu svrhu se koristi RAM 7. Jednostavno, RAM 0 se ukloni iz vrha memorijske mape i tu se dovede RAM 7. Naravno, čim korisnik napusti ekranski editor, RAM 0 će se vratiti na svoje mesto, bez gubitka sadržaja. Sva ova preklapanja vrši sam operativni sistem bez bilo kakve intervencije korisnika.

Ram — Tape

Preostale su još stranice 1, 3, 4 i 6, koje, bar što se tiče operativnog sistema, kao da i ne postoje. Sinkler je tih 64K zamislio kao



„ram-disc", tj. kao medijum za brzo odlaganje i učitavanje podataka. Bolji naziv bi, međutim, bio „ram-tape", jer je pristup isključivo sekvencijalan: bajtovi se „snimaju" kao na kasetu, jedan za drugim, a onda se u istom tom redosledu mogu i čitati. Sve to, razume se, ekstremnom brzinom.

Zanimljivo je da se operacije sa ram-diskom obavljaju uobičajenim bejzik naredbama **SAVE** i **LOAD** (**VERIFY** je, jasno, nepotrebna). Korisnik jedino mora da otvori odgovarajući kanal za komunikaciju.

Ula — diktator

Jedna od koristi ram-diska bila bi čuvanje sadržaja ekrana i brzo smenjivanje slike. To „brzo", međutim, i nije tako brzo (u svakom slučaju ne trenutno). Bolje je tada zaobići naredbe **SAVE i LOAD**, i preslikavati blokove bajtova mašinskom naredbom **LDIR**.

Ipak postoji i specijalan način zamene slike, dostupan jedino iz mašinskog jezika. Setovanjem posebnog indikatora ULA se može navesti da čita sliku ne sa uobičajene adrese „4000, već iz najvišeg bloka memorije, tamo gde spada RAM 7. To je zaista trenutna operacija, ali krije u sebi i malu zamku. Vlasnici „spektruma" svakako znaju da ULA ne dopušta procesoru pristup bloku video memorije u fazi čitanja slike. Z80 je tu doslovce nemoćan, jer mu ULA prosto prekine klok i tako ga efikasno onesposobi za neko vreme. To dosta usporava sve programe koji se izvršavaju iz najnižeg dela RAM-a u verzijama „spektrum" i „spektrum +". Sada, na „spektrumu 128", ULA „teroriše" i na višim adresama, što i nije baš najprivlačnije rešenje.

Čini se da je ipak najjača strana „spektruma 128" njegov muzički čip. Kad programeru dojade RAM-ovi, ROM-ovi i njihova „organizacija", može potražiti mir uz zvuke koje uspe da isprogramira. Bejzik mu, naravno, ni u tu svrhu neće biti od bog zna kakve koristi.

Jovan Skuljan

stolari i računari

Ko se boji domaće elektronske industrije?

Nakon naše vesti o proizvodnji serije računara „sperry PC 500" u „Računarima 12", u domaćoj medijskoj ali zaredala su mišljenja o nepotrebnim i potrebnim pokušajima „šrafciger" industrije. To je saradniku „Računara" Branku Habrangu dalo povoda da se još jednom oglasi na istu temu.

Iznenađujuće zanimanje domaćih medija izazvala je vijest o uključivanju Drvne industrije „Gaj" iz Podravske Slatine u tokove elektronske industrije. Pojedince je najviše začudilo sklapanje prve serije od 64 računala marke „sperry PC 500" u novoosnovanoj osnovnoj organizaciji udruženog rada „Elektronika — Gaj", a u suradnji sa „Sperry Univacom" i zagrebačkim „Infosistemom", o čemu su pisali neki listovi, a jugoslavenska javnost obavještena je putem naše revije „Računari" broj 12. Taj posljednji naslov, čini se, pobudio je posebnu pažnju novinskih izvjestitelja, pa je „Gaj" završio na novinskim stupcima glasila širom zemlje, a ni televizija nije u tome zaostala.

Zabrinuti čipobrižnici

„Stolari prave kompjutore", zaključili su u jednom visokonakladnom listu čudeći se kako je to moguće, zabrinuti za budućnost dotične tvornice, ali i domaće elektronske industrije. Bilo je to tipično reagiranje u nedostatku potpunijih informacija o mogućnostima proizvodnje računala, informatičke opreme i profesionalne elektronike u nas. Da su informatički znatiželjnici okrenuli list više u „Računarima 12", mogli su pročitati obrazloženje kibernetičara Branislava Bakića o mogućnostima proizvodnje računala u zemlji. Dakle, Bakić je ustvrdio da nam na raspolaganju stoje četiri načina: vlastiti razvoj, uključivanje u OEM tržište, kooperacija i licencna proizvodnja. Pri tome se rad s originalnim elektroničkim sklopovima (Original Equipment Manufacturer) kod nas poistovjećuje sa „šrafciger" industrijom.

Budući da kasnimo desetak godina za zbivanjima na tržištu malih računala, danas ne možemo puno birati kako ćemo razvijati domaću proizvodnju računala. Postoje dva tri veća proizvođača koja su krairala vlastiti razvoj, no ni oni nisu usvojili masovnu proizvodnju, akamoli tržište. Za to postoje razlozi. Uz te veće proizvođače razvijali su se manji koji su se postepeno uključivali u svijet elektronika. Neki od njih će se održati, neki će ubrzo proizvodnju preorijentirati na nešto drugo. Slično je učinio i „Gaj", otvarajući vrata tržišta uz suradnju iskusnijih partnera, ali ne samo u proizvodnji računala. Primjerice, sada su u suradnji s partnerom iz NR Mađarske proizveli registracijske magnetofone, uređaje za zapisivanja telefonskih razgovora s nekoliko kombinacija ulaza i izlaza.

Jasno je da „Gaj" prvenstveno i dalje ostaje prerađivač drveta, među najvećima je u Hrvatskoj, a u proizvodnji namještaja koristi i NCR strojeve, čime se mogu podičiti rijetki domaći prerađivači drveta. U „Ekonomskoj politici" broj 1772 Dragiša Bošković pišući o dilemama u domaćim medijima zabilježio je: „Zato se ovde od sveg srca daje glas za „Gaj" iz Podravske Slatine kao ponuđača računara i unapred obećava da će taj glas biti dat svakoma ko nađe načina da od dobavljenih delova nekako sklopi kompjuter i doturi ga jugoslavenskom kupcu. Činimo to naravno zato što nemamo mnogo izglada da se problem reši kao što bi ga rešila zemlja doista usmerena da se preko dinara što slobodnije pretvorljivog u svetski novac i uz normalnu zaštitu na granici otvori prema svetu. Zato ostaje samo taj okolišni put umnožavanja stotine, poželjno hiljada najraznovrsnijih ponuđača kompjutera koji kreću u posao željni zarade ali u međusobnom nadmetanju obaraju cene. Doslovno tako nešto uostalom upravo se zbiva u Indiji, na čijem je tržištu slično nadmetanje prepolovilo prošlogodišnje cene."



Bura u čaši vode: Sklopovi s OEM tržišta u podravskom „Gaju" koriste se i za proizvodnju profesionalne elektronike

Kad se slegne prašina

Kad se malo obriše podignuta prašina oko „stolara i računala" otkriva se istina da je osnovni cilj ipak kompjutorizacija proizvodnje. Tržišna ekspanzija malih računala treba podići informacijsko znanje na višu razinu kako ne bi trebali današnji školarci u tvornici još jednom učiti kako treba raditi. Dobro organizirana poduzeća lakše održavaju korak s konkurencijom, ulažu u kadar i razvoj i s većom vjerojatnošću će uspješno organizirati proizvodnju i u 21. stoljeću.

Ilustrativno je da je 1972. godine na savjetovanju Informatika u SR Hrvatskoj i njezin daljnji razvoj konstatirano da treba pojačati ulaganja na tom području ako ne želimo još više zaostati. Onda je istaknuto da u svijetu u organizaciji znanstvenih istraživanja i razvojnih djelatnosti troškovi za informaciono-dokumentaciono--komunikacijski sistem iznose 25 posto troškova od ukupnih ulaganja u istraživanje i razvoj. Kod nas se tada za te potrebe trošilo 50 puta manje.

Pet godina kasnije u „Impulsu", poslovno-interesnoj zajednici proizvođača telekomunikacija, elektronike, kabela i računara utvrdili su da su ulagana znatna sredstva kroz proviziju u razvoj tržišta opreme stranih proizvođača, dok je podrška domaćoj industriji gotovo u potpunosti

izostaia Domaći proizvođači su za uvoz repromaterijala plaćali visoka carina a strani proizvođači imali su pogodniji carinski režim Uvoz repromaterijala pored deviznih teškoća bio ja znatno otežan ili onamogućan zbog narazjašnjanih pojmova što je uopće rapromaterijal za proizvodnju oprema za automatsku obradu podataka pri čemu ja uvoz sklopova i tunkcionalnih jedinica koja i strani proizvođači nabavijaju na OEM tržištu bio praktički onamogućen U takvim okolnostima domaći proizvođač ja na domacam tržištu bio u napovoljnijem položaju od stranih proizvođača što nigdje u svijetu nija slučaj zapisano ja tada u jednoj informaciji Impulsa

U to ja vrijama samo za četiri godina uvezana oprama za automatsku obardu podataka vrijedna 220 milijuna dolara ondašnjih četiri milijarda dinara U Impulsu' su procijanili da smo za proviziju tada dali 1.2 do 1 6 milijardi dinara, a da smo domaće tržište praktično besplatno poklonili stranim proizvođačima

Takva situacija na tržištu velikih kompjutorskih sistema nija mogla biti baza za uključivanje zamlje u svijat informatika pa smo danas potpuno inferiorni na tržištu mikroelaktronika

Ipak se kreće

Velika računala ipak su nakako ulazila u ureda u vrijeme investicijskog zamaha a kada ja prevledala minijaturizacija u elektronskoj industriji i kada je došlo vrijema malih računala demokratizacija informatičkog osnovnog sredstva edministracija ja mala računala svrstala u red — luksuzna robe Zakonom o privremenoj zabrani korištenja društvenih sredstava za financiranje neproizvodnih i naprivrednih investicija, informatička je oprama ubrojana u administrativno-birokratsku uredsku opremu To ja onemogućilo nabavu te oprama razvitak kompjutorske industrija pa ja počelo razdoblje šverca malih računala koja i danas traja Tak promjenom propisa, potkraj 1984 godina dozvoljan ja uvoz malih računala s vrijednosnim ograničenjam u početku od 40 000 dinara poslije 60 000 a sada 90 000 dinara Uvoz oddatna opreme pri tome nija bio posebno reguiiran

Zatvorenost tržišta odrazila se i na školski sustav koji nija na vrijema reagirao i uveo informatiku kao obvezatni predmet Luk i pilica i danas su još didaktičko sredstvo osnovnoškolaca a kompjutorsko obrazovanja zaposlanih u privredi nija ni u začetku

No i u takvim uvjetima razvila se kakva-takva produkcija računala Spomenut ćemo najpoznatija domaće proizvođače

Prvi su 1981 godina s računalima krenuli u varaždinskom PEL-u i beogradskom „Ivi Lola Ribaru PEL ja u konačnici napravio boiji posao i njegova računala tipa „orao prihvaćena su za korištenja u školama u SR Hrvatskoj Pravi boom na tržištu svojevremano ja napravljan prvim domaćim računalom u kitu s „galaksijom koja ja dosad postigla prema raspoloživim podacima, najveću seriju Školski servis Iz Zagraba proizvodi malo računalo univerzus kompatibilno s „appla II a „Ivasim-Elaktronika iz Ivanić-Grada prodaja računalo ultra koja naziva univerzalnim kompatibilnim računalom „iskra uz poslovna sisteme proizvodi i malo računalo HR-84 dok „Javor' trgovinska radna organizacija OOUR Informatika i elaktronika Bitolj proizvodi računalo marta Novosadski „Novokabel proizvodi kompjutor ERA a zagrebački tehničar tera Ima i drugih proizvođača sve do privatnih ukupno dvadesetak Uz njihovu produkciju domaće tržište obogaćeno ja i s nakoliko tipova računala proizvedenih u kooperaciji sa stranim tvrtkama a putem zastupništva stranih proizvođača u nas se mogu kupiti i neka računala s konsignacije inače svjetska ponuda malih računala obuhvaća više od 300 modela

Ponuda oprema za računala već ja slabija a programska podrška ujedno je i najslabija točka svih domaćih proizvođača Zaokupijani usvajanjem hardvera tizičkih dijelova računala, zapostavili su programsku podršku pa poduzeća mogu kupiti mala sisteme ali na i programa za određene djelatnosti Pri svemu tome treba znati da je naša industrija informatička oprema među najmlađima u zemlji a da su fundamentalna istraživanja na tom području vremanski dugotrajna i vrlo skupa To ja samo jedna dimenzija tog složenog problema

Razbiti žabokrečinu

U Jugoslaviji zbog poznatih privrednih teškoća sve se manje ulaže u razvoj Tako se 1976 godina za razvoj odvajalo 1 07 posto nacionalnog dohotka, a 1983 samo 0 91 posto Znanstveni radnici u takvim prilikama nisu ni dovoljno plaćeni zato se bave sivom ekonomijom projektima i istraživanjima za honorar To s razvojem nama nikakva veza

U domaćim privrednim razvojnim jedinicama ima desat posto u Slovaniji 20 posto istraživača U susjednoj Italiji ima 40 posto a u SAD-u 70 posto od istraživača spomenutih zamalja

Takva kretanja zavisna su od stupnja razvoja privrede i strukture faktora proizvodnja U zanatskom načinu proizvodnje rad ja zastupljan sa 70 posto kapital s 20 a znanja sa skromnih desat posto U industrijskoj proizvodnji rad je smanjen na 25 posto udjel kapitala povećan ja na 50 posto a znanja na 25 U informacijskom društvu društvu sutrašnjice, rad će sudjelovati s pet do desat posto kapital s 25 posto dok će ostatak zauzimati znanja, informacija U takvom postindustrijskom društvu čovjak bi trabao živjati ugodnija, s više slobodnog vremana za zabavu rekreaciju učenja

Za takvu budućnost vjerojatno danas u nas na postoji dovoljan fond znanja Na samo u privradi već ni na visokoškolskim institucijama i institutima Dok se na stvori takva kritična masa znanja — učenjam istraživanjima transferom i usvajanjem tuđih spoznaja i iskustava — neće se moći ni organizirati proizvodnja s manjim udjalom rada Proizvodi iz takvih tvornica teško će nalaziti kupca

Treba unijati namir u naš mir stagnacija i žabokračina uskih horizonata bavljenja samim sobom i raspodjelom postojećeg svojavrsnog zatvaranja očiju i fatalizma pred budućnosti zapisao ja dr Drago Gorupić pišući o znanstvano tahničkoj revoluciji i zadacima poalovodnog osoblja Traba se otvoriti budućnosti traba nam približiti budućnost informacijski analitički i sukobom mišljenja da bismo napokon počeli o njoj misliti i za nju se pripramati Samo to traba činiti brzo jer nas svijet neće čakati

Branko Hebrang

U sledećem broju

Novi projekat

TURBO DOS

Prvi nastavak kompletnog hard versko-softverskog projekta disk interfejsa za spektrum sa paralelnim interfejsom za štampač i džojstik i monitorskim izlazom Interfejs je opremljen i sopstvenim ROM-om od 8 K koji radi „u senci' osnovnog ROM-a i potpuno je kompatibilan sa mikrodrajvom Program od 32 K TURBO DOS učitava za samo sedam sekundi

Nova serija

POLIKLINIKA C

Stručni saradnick Računara iz Njujorka prof dr Radomir A Mihajlović pripremio je opsežnu školu jezika C — polikliniku u kojoj će bolesnika od bejzika paskala i fortrana pokušati da „izleči ovim sve popularnijim programskim jezikom koji de facto postoja svetski standard

Umetak

Dejan Ristanović

ŠTAMPAČI

Detaljan priručnik za upotrebu programiranje i adaptaciju matričnih štampača na YU standarda

LOTO NA RAČUNARU

Žarko Vukosavljević, ekspert za loto sisteme pripremio je za vlasnike spektruma program koji im utorkom uveče sigurno neće donositi miliona ali će im pomoći da racionalno ulažu svoj novac i igraju skraćene sisteme samo sa svojim srećnim brojevima

HAKERSKI VODIČ MINHENA

Početkom maja Zoran Životić obišao je sve poznatije prodavce IBM PC/XT klonova u pojasu između Minhena i Frankfurta Adrese izbor cena uslovi nabavke i konačno kako na PC reaguju YU carinici



Dejan Ristanović

Dejanove pitalice

Svi igraju Loto

Tvrdnju Jugoslovenska lutrija da je Loto sve popularniji ilustruje i naš šesti nagradni zadatak, koji je izazvao za sada najveće interesovanje; u predviđenom roku smo primili 327 odgovora, od toga svega 14 pogrešnih. Iako je ogromna većina rešavača koristila računar, nagrade su zaslužili čitaoci koji su pomalo zaposlili i svoj mozak!

Podsetićemo se kao i obično najpre zadatka trebalo je pronaći Loto kombinaciju čija je oznaka 10 000 000 (10 miliona) kao i oznaku kombinacije 5 8 10, 11 17 25 26 Zadatak se može rešiti prilično jednostavno primenom takozvane grube sile sedam koncentričnih FOR-NEXT petlji koja prebrojavaju kombinacije će sa strane programera biti sasvim dovoljne Ovako je rešenje sa druge strane *vrlo* neprijatno za računar ako program napišete na bejziku spektrum će ga žvakati četiri do pet dana! Rešenja dobijana na ovaj način nas navode da ubuduće na kraju ove rubrike pišemo *redakcija Računara ne snosi odgovornost za kvarove nastale u toku rešavanja ovoga zadatka*

Neki su se rešavači dobro dosetili da upotrebe jedan od popularnih bejzik kompajlera na taj je način vreme izvršavanja smanjeno neke 3—4 puta ali je i dalje predugačko Bolji programeri koji su operisani od matematike pisali su program na asembleru dobijajući tako rutine koje daju rezultat prihvatljivo brzo — za svega par sati Do rešenja se međutim može doći za isto toliko sekundi primenom programa poput onoga koji objavljujemo

Kako bismo ovaj zadatak rešili bez primene računara? Naučićemo najpre da izračunamo ukupan broj kombinacija u igri loto sedam se brojeva može izdvojiti iz 39 na 39 nad 7 načina N nad K' računamo iz formule

$$\binom{N}{K} = \frac{N*(N-1)* \ldots *(N-K+1)}{1*2*3*4* \ldots *(K-1)*K}$$

Da vidimo koliko od ovih kombinacija počinje brojem 1 ovo se izračunavanje

```
10 REM          L O T O
20 REM
30 REM Prema programu Zoltana Romana
40 REM
50 REM         "Racunari 15
60 REM
70 DIM A(10)
80 CLS
90 PRINT "1  Oznaka po kombinaciji"
100 PRINT "2  Kombinacija po oznaci"
110 PRINT
120 INPUT A
130 ON A GOSUB 310,170 ELSE END
140 END
150 :
160 REM Kombinacija po oznaci
170 INPUT "Oznaka "S: M=39:G=0
180 FOR W=6 TO 1 STEP -1
190 H=7-W
200 G=G+1:Q=M-1
210 GOSUB 400
220 IF T<S THEN S=S-T:M=M-1:GOTO 200
230 M=M-1:A(H)=G
240 NEXT W:A(7)=G+S
250 PRINT "KOMBINACIJA: ";
260 FOR H=1 TO 7:PRINT A(H);: NEXT
270 PRINT
280 RETURN
290
300 REM Oznaka po kombinaciji
310 S=15380937
320 INPUT "Kombinacija "A(1),A(2),A(3),
    A(4),A(5),A(6),A(7)
330 M=39:W=7
340 FOR J=1 TO 7
350 Q=M-A(J):GOSUB 400:W=W-1:S=S-T
360 NEXT
370 PRINT "Oznaka za ovu kombinaciju
    je ";S
380 RETURN
390 :
400 IF Q<W THEN T=0:RETURN
410 T=1:IF Q=W THEN RETURN
420 R=Q-W:K=1:IF R<W THEN R=W
430 FOR I=R+1 TO Q:T=T*I/K:K=K+1:NEXT
440 RETURN
```

svodi na pitanje koliko se kombinacija može napraviti izborom 6 brojeva od 38 tj na 38 nad 6 načina Obzirom da je 38 nad 6 jednako 2 760 681 što je dosta manje od 10 miliona kombinacija sa traženom oznakom ne počinje jedinicom Ako bismo pretpostavili da kombinacija počinje dvojkom njen bi redni broj mogao da bude najviše 37 nad 6 ili 2 324 784 Nastavljajući sličan postupak nalazimo da je 35 nad 6 jednako 1 623 160 što sabrano sa 2 324 784 i 1.947 792 daje 8 656 417, broj koji je manji od 10 miliona Ako bismo međutim pretpostavili da dobitna kombinacija počinje šesticom ovom bismo zbiru dodali 34 nad 6 tj 1 344 904 dobivši 10 001 321 broj veći od 10 miliona Tražena kombinacija dakle počinje brojem 5! Ponavljajući sličan postupak dobijamo da kombinacija sa oznakom 10 000 000 glasi *5 27, 29 30 34 35 36* što je ujedno i rešenje prvog dela zadatka

Drugi deo zadatka može da se reši nešto brže bez mnogo probanja Kombinacija koje počinju brojevima 1 2 3 i 4 ima 39 nad 7' — ,35 nad 7 ili 8 656 417 (ovaj smo broj več dobili doduše na nešto drugačiji način) Zatim računamo koliko ima kombinacija koja počinju sa 5 dok im je drugi broj 6 ili 7, dobijamo ,34 nad 6 — 32 nad 6 ili 436 712 Ponavljajući ovakav postupak i sabirajući međurezultate dobijamo da oznaka kombinacije 5 8 10 11 17 25, 26 glasi *9124166*

Pažljivo pregledavši prispele programe pokušali smo da izaberemo najkraći i dodelimo mu prvu nagradu U najužoj konkurenciji su se našla rešenja *Jelene Grujić* iz Subotice i *Romana Zoltana* iz Zrenjanina Obzirom da je program drugarice Grujić bio nešto kraći otkucali smo ga i detaljno testirali primetivši mnogo sitnica koje bi bile interesantne za profesora Slavića i njegovu rubriku To može i bolje iako je program tačno generisao kombinaciju sa oznakom 10 miliona kombinacija sa oznakom 1, 2 3 30 su bile jednake zbog

Jedan programerski problem

*Osmi nagradni zadatak ispunjava zahtev čitalaca koji stalno traže čisto programerske probleme Promenljivima A B C D i E su dodeljene vrednosti koje treba sortirati Program koji će obaviti takvo sortiranje **ne sme** da sadrži cikluse (eksplicitne ili implicitne) GOTO naredbe koje pokazuju „unazad i bilo kakve strukture **osim** IF THEN ELSE naredbi Ne smeju se osim toga dimenzionisati nikakvi nizovi niti koristiti skalarne promenljive **osim** pomoćne promenljive Z Samo se po sebi razume da čitav program treba da bude napisan na nekom višem jeziku (na primer bejziku ili paskalu) i da ne sme koristiti naredbe PEEK POKE i USR niti specifičnosti računara*

*Pošto napišete program prebrojte IF naredbe i upišite njihov broj u kupon koji dajemo na kraju ove stranice Kupon zajedno sa programom pošaljite na adresu „Računari (za Dejanove pitalice), Bulevar vojvode Mišića 17 Beograd tako da pristigne pre **15. juna 1985.** Među programima koji rešavaju traženi problem primenom minimalnog broja IF naredbi će biti izvučena prva (10.000) druga (5 000) i treća (3 000) novčana nagrada*

*Ispunićemo još jedan zahtev naših čitalaca nije više neophodno seći originalne kupone Ukoliko ne želite da oštetite svoj primerak „Računara" prepišite **kompletan** kupon na početak prve strane rešenja tako da bude uokviren i jasno uočljiv*

kumulativne računske greške N nad K je, uz ostalo računato na numerički neprihvatljiv način! Tako je *prva* nagrada u iznosu od 10 000 dinara pripala *Žoltanu Romanu* čiji (nešto prerađen) program dajemo na slici

Preostale tačne odgovore smo ubacili u tradicionalni koverat iz koga smo izvukli pisma *Bojane Stojanovića* iz Smedereva i *Ivana Vrbovčena* iz Kapele kojima su respektivno pripale druge (5 000) i treće (3 000 dinara) nagrade

```
Pet skalara moze da se sortira primenom
najmanje ____________ IF naredbi

Ime i prezime ______________________________

Adresa        ______________________________

Mesto         ______________________________
```

svi „atarijevi" programi

Na tržištu hardvera i softvera, zahvaljujući računarima iz serije „atari st", poslednjih meseci vlada neuobičajena živost. Novi naslovi uslužnih, sistemskih i jezičkih programskih paketa izlaze doslovce svakoga dana i „atari" biblioteka je već narasla na gotovo 500 programa. Uz niz noviteta iz „atari" tabora, „Računari", verovatno prvi u svetu, objavljuju spisak svih „atari st" programa prema podacima koji su nam bili dostupni do sredine aprila.

Operacijski sistem TOS računala serije atari st kao što je poznato nije se dosad isporučivao u ROM-u nego na tzv sistematskoj disketi čiji sadržaj je nakon svakog uključenja računala trebalo upisati u RAM memoriju Sama firma Atari je ovaj potez objašnjavala potrebom da se OS prije „ugradnje u ROM dobro istestira u praksi Zlobnici (čitaj konkurencija s la Commodoraj su pak tvrdili da Atari ima problema zapravo što u operacijskom sistemu koristi grafički sistem OEM firme Digital Research Ovaj grafički sistem je po vanjskom izgledu (ali ne i po zastavu) vrlo sličan grafičkom sistemu koji koristi firma Apple na svojem računalu makintoš Pronala su se glasina da je Apple zbog toga tužio Digital Research sudu i da se — pošto je vjerojatno da će tužbu i dobiti — TOS, koji koristi GEM neće moći koristiti na računalima serije ST



Mali lopovi — velika dobit

Istina je vjerojatno negdje u sredini Točno je da je firma Apple prigovorila Digital Research-u zbog vanjske sličnosti GEM-a i njihovog makintoša ali pošto i sama firma Apple namjerava kod Digital Research-a (kao najvećeg proizvođača sistemskih programa za mikrokompjutera koji je stvorio na primjer i CP/M) naručiti neke sistematske programe za svoja nova tipove računala vjerojatno je ocijenila da nije uputno baš previše gnjaviti novog partnera Tako su se Apple i Digital Research na kraju i bez suda dogovorili da se ipak izvrše neki kozmetički zahvati na OEM u i ovaj više ne buda tako napadno sličan grafičkom sistemu makintoša

Ovakav postupak firma Apple uzbudio je američku (i svjetsku) scenu mikroračunala (pa čak i šire uključivo čitavu elektroniku i tehniku općenito) Naime silan razvoj elektronike a naročito mikroračunala posljednjih godina bio je moguć zbog niza malih poodljšanja i unapredanja koja su razne konkurantske firme ugrađivale u svoje proizvode na bi ii se nekako razlikovali od susjedovih i time lakše reklamirali i prodavali Jasno je da svaka mala firma ne može svake godine smisliti neki novi revolucionarni proizvod kojim će zasjeniti svoje konkurente i ostvariti dovoljan prihod da financira razvoj slijedećeg novog revolucionarnog proizvoda Tako im ne preostaje ništa drugo nego da proizvodima svojih konkuranata dodaju neka mala izmjena i poboljšanja Ovakav razvoj malim koracima je prešutno prihvaćen čak i u takvom društvu kao što je američko koje je i zasnovano na što većem profitu iz pojedinih proizvoda Ako je i bilo nekih očitih kopiranja nikome se nije isplatilo da rješava stvari sudom zbog vrlo sporog kompliciranog i skupog američkog sudskog sistema

Međutim sada je prvi puta neka velika firma (a Apple to jeste) sa velikim financijskim zaleđem najavila da želi zaštititi ne samo svoje sistemske programe nego i sam njihov vizuelni izgled To je izazvalo uzbunu među mnogobrojnim malim firmama koje su već užurbano radile na nekim poboljšanjima i sitnim izmjenama svojih grafičkih sistema Svaka od njih se našla u situaciji da dobro promisli ima li smisla ulagati znatna financijska sredstva u razvoj sistemskih programa za koja više nije sigurno da li će se moći i smjeti prodavati Veliki broj takvih firmi na kojima u stvari i počiva snaga razvoja američke tehnologije počeo je javno izražavati neslaganje s takvom politikom firme Apple pa čak i otkazivati poslova s njom Također u mnogim stručnim časopisima pojavljivali su se uvodnici i pisma čitalaca koji su zvonili na uzbunu za spas inventivnog duha američke tehnologije , „za ugroženi razvoj tehnologije i općeniti prosperitet društva itd Neki od njih su iščeprkali i podatak da je i grafički sistem makintoša u stvari kopija grafičkog sistema razvijenog u laboratorijama firme XEROX

Vjerojatno je i to utjecalo na odluku firme Apple da izgladi spor s Digital Research-om Iako pojedinosti njihovog dogovora nisu sasvim poznate široj javnosti izgleda da je Apple ipak uspio prisiliti Digital Research da poduzme određene kozmetičke zahvate na OEM-u Međutim kao što i svaka medalja ima dvije strane Digital Research tvrdi da te izmjene u OEM-u ne moraju nužno biti korak unatrag u kvaliteti i korisničkoj prihvatljivosti programa Navodno sada su im širom otvorene vrata da znatno prepravi i poboljšaju GEM i učine ga stvarno najboljim grafičkim sistemom — kao što su i reklamirali pri njegovom najavljivanju

Najzad u ROM-u

Neupućeni čitalac će odmah pomisliti da svakav razvoj događaja mora imati posljedice i za firmu Atari, koja koristi upravo spornu verziju GEM-a na svojim računalima serije ST Međutim firma Atari hladnokrvno izjavljuje da je od Digital Research-a kupila (i platila) GEM i da je se dalje ne tiču i ne obavezuju nikakvi dogovori (ili problemi) Digital Research-a s trećim licima. Da potkrijepi takav svoj hladnokrvni stav upravo ovih dana pustila je u prodaju i ROM-verziju operativnog sistema TOS koja sadrži originalnu neizmjenjeni OEM ROM-verzija se prodaje u kompletu od 9 promova (Programmable Read Only Memory) po 32 K ukupno 192 K Cijena je zajedno s besplatnom montažom u prodavnici Atari računala 29 funti ili 99 DM

To je smiješno niska cijena čak manja nego cijena odgovarajućeg broja eproma koja su neki od korisnika računala serije ST već isprogramirali i ubacili u prazna podnožje predviđena za ROM-memoriju s operacijskim sistemom Ovakav potez firme Atari naišao je na nepodijeljene simpatije svih korisnika računala serije ST od kojih su neki bili i ogorčeni zbog toga što su kupili 520 ST po cijeni po kojoj se tri mjeseca kasnije prodavao 520 ST+

Ugradnjom operacijskog sistema u ROM-u računala tipa 260 ST imaju na raspolaganju za programe i podatke oko 475 K RAM memorije pošto se od sveukupnih 512 K mora odbiti 32 K za video-memoriju i oko 2 K za sistemska varijabla operacijskog sistema To vrijedi i za dosad prodata računala tipa 520 ST koja se više ne proizvode Njih je zamijenio model 520 ST+ kod kojeg već i prije nije bilo problema s memorijom budući da ima ugrađenih čitavih 1 MB (tj 1024 K) RAM-a.

Zbog teškoća u proizvodnji i isporuci svojih kolor-monitora SM1224 firma Atari prihvatila je ponudu francuskog poduzeća Thomson i u svojim kolor-kompletima prodaje njihov kolor monitor CM-3638ZAR On je dodušo za oko 10% skuplji od originalnog Atari monitora, ali zato ima i nešto veću dijagonalu ekrana oko 36 cm (14) Neki trgovci u kompletu prodaju

kolor-monitor SONY KX-14CP 1 koji je još nešto skuplji. Kvaliteta kolor-slike na svim kolor-monitorima je vrlo dobra, ali je opće uvjerenje da je za poslovne primjene naročito obradu teksta ili podataka mnogo prikladniji standardni Atari crno-bijeli monitor SM124

Još jedan model

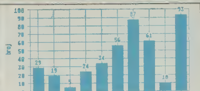
U Americi je nedavno otpočela prodaja novog modela računala iz serije ST — modela 1040ST. Taj model također, ima ugrađen 1 MB RAM memorije ali i već ugrađenu disk-jedinicu kapaciteta 720 K formatirano i ugrađeni ispravljač. Operacijski sistem TOS je razume se u ROM-u. Predviđena cijena novog modela zajedno s crno-bijelim monitorom visoke razlučljivosti iznosi 999$ ili oko 3000 DM. Prema toj informaciji, ne planiraju se novi modeli serije ST s još većom radnom memorijom pošto je dostignuta gornja granica raširenja RAM memorije trenutno raspoloživim (jeftinim) memorijskim čipovima na tržištu

Ali već za sredinu ove godine najavljena je prodaja dodatnog modula za računala serije ST, koji će se s računalom povezivati preko DMA priključka (direct memory access). Dodatni modul će imati ugrađen procesor MC68020 i 2MB RAM-a (proširljiva do 4 MB). Kao što je poznato procesor MC68020 smatra se jednim od najjačih mikroprocesora na svijetu i ima potpunu 32-bitnu arhitekturu (i adresu sabirnicu i sabirnicu podataka). Ovaj dodatni modul će omogućiti računalima serije ST da rade s operacijskim sistemom UNIX — najjačim operacijskim sistemom na mikro i miniračunalima za više korisnika i više poslova istovremeno (multi-user multitasking). Pošto za operacijski sistem UNIX postoji mnoštvo odličnih poslovnih i komercijalnih programa ova najava se smatra za objavu rata firmama koje trenutno kontroliraju tržište malih poslovnih računala (uglavnom IBM i DEC). Očito firma Atari nastoji razbiti tamu da je neozbiljna firma koja se uglavnom bavi proizvodnjom neozbiljnih računala za igranje

U izravnoj vezi s mogućnošću rada više korisnika i više poslova istovremeno je i potreba za tzv. tvrdim diskom tj. disk-jedinicom velikog kapaciteta i velike brzine prenosa podataka. Iako s proizvodnjom tih disk-jedinica firma Atari ima određenih problema, već je isporučeno stotinjak komada i to uglavnom velikim programskim kućama koje grozničavo prepravljaju svoje poslovne programe da mogu koristiti i tvrdi disk. Ovaj model još nema službenu oznaku a ima kapacitet od 10 MB. Njegova najavljena cijena je 600 funti ili 2000 DM

Javno dobro

Bilo je nagađanje da programski jezik bejzik isporučivan zajedno s računalima serije ST a poznat po svojoj glomaznosti nespretnosti i sporosti nije konačna verzija. Firma Atari je ova nagađanja demantirala izjavivši da sama ne misli raditi na daljim poboljšanjima bejzika. Međutim, već je najavljeno da će firma Metacomco ubrzo izbaciti na tržište svoj bejzik kompajler za ST. To će vjerojatno, obradovati sve one kojima je bejzik drugi materinji jezik, pošto je firma Metacomco u posljednje vrijeme poznata po vrlo dobrim programima a i vrlo usko surađuje s firmom Microsoft, tako da će vjerojatno njihov bejzik kompajler podržavati sintaksu (de fakto standard) Microsoft. Prema najavi Metacomco



Izjednačena snaga iz dijagrama (dobijenih pomoću programa VIP Professional) ne može se izvući pouzdani zaključak koja je vrsta tržište prihvatila „atari st"

bejzik kompajler će se prodavati i u ROM-u (128 K ROM-modul)

Od početka godine uz računala serije ST više se ne isporučuje programski jezik Dr Logo nego nova verzija SPEED-LOGO. Ova verzija je navodno oko tri puta brža od prijašnje. Svi vlasnici

ATARI-ST	programi
grafika	29
komunikacije	19
muzički	5
obrada podataka	24
obrada teksta	34
pomoćni	56
poslovni	87
programski jezik	61
obrazovanje	10
zabava	93
ukupno	418

računala serije ST mogu ovu verziju dobiti besplatno kod svojeg prodavača. Time se proširuje već ionako za računala takve klase i cijene prilično široka paleta tzv. public domain software tj. programa koji su namijenjeni širokoj javnosti i besplatnom presnimavanju i širenju. Osim programskih jezika bejzik Dr Logo i Speed-Logo tu su još i grafički programi GEM-DRAW Neochrome i DrDoodle tekst-procesor GEM-WRITE i neki od programa iz razvojnog sistema (Command i Kermit). Među te besplatne programe spada i tzv. „Amiga-Ball" tj. program koji na kolor-monitoru računala serije ST imitira poznatu skakutajuću loptu reklamnih demo-programa firme Commodore za njihovo novo računalo „amiga". Zlobnici kažu da je vjernost i prirodnost odje i pokreta lopte mnogo bolja na računalima ST nego na originalu „amigi". Ali možda najznačajniji od svih public domain programa je CP/M-2.2 emulator program koji otvara računalima serije ST vrata u svijet velikog broja CP/M programa (više o tome u prilogu „Atarijev povratak u budućnost")

Umjesto programskog paketa GEM-Draw+GEM-Write koji Digital Research nije uspio dovršiti na vrijeme (u najavljenom obliku i kvaliteti) firma Atari sada isporučuje programski paket koji se sastoji iz programa First-Wort (tekst-procesor) DEGAS (grafički program) i DB-Master (database)

Programa kao pleve

Firma RDS-Software nudi komplet programa namijenjenih za sistemsko programiranje u programskom jeziku C. U kompletu su osim samog C-compilera koji je usput rečeno i združljiv (kompatibilan) s UNIX-sintaksom a na izlazu daje već optimizirani program u strojnom kodu još i C-Link tzv. parser koji provjerava sintaksu izvornih programa u jeziku C i omogućava ispis unakrsnih referentnih listi (cross reference list). Tu su još i tri puna disketa knjižnica (library) raznih C-funkcija kao i C-Archive program za pretraživanje knjižica funkcija kao i stvaranje vlastitih knjižnica funkcija iako ne u izravnoj vezi s programiranjem u jeziku C u kompletu su još i urednik sličica oblika slova i aktivnost „miše" (icon font i mouse editor). Neovisno o programskom paketu može se dobiti i GEM Disk-monitor koji omogućuje šaranje po sadržaju disketa (pirati dobro znaju što se time sve može postići) kao i Proficopy za brzo presnimavanje sadržaja disketa, koji čak i kod 260ST presnimava čitav sadržaj disketa u jednom potezu

Već spomenuta firma Metacomco upravo je dovršila Lattice-C-compiler čija su knjižnice funkcija napravljene po uzoru na UNIX-ove. Već otprije je poznat njihov paskal kompajler kao i razvojni paket za programiranje u strojnom jeziku (Assembler Development Kit). Upravo se dovršava i kompajler za programski jezik LISP (List Processing Language) koji se mnogo koristi u razvoju umjetne inteligencije (AI = Artificial Intelligence)

Njemačka firma Kingsoft (poznata po šah-programu Grandmaster za C-64) izdala je i prvu njemačku zabavnu igru za čitavu porodicu. Program postavlja razna tekstovna ili grafička pitanja i pritom svojim odgovorima zabavlja do 15 učesnika

PROGRAMSKI JEZICI

4xFORTH	The Dragon Group
Assembler/Monitor	Computer One
BASIC Interpreter	Crupe GmbH
BASIC Interpreter	Philon
C Compiler	Computer One
C Compiler	GST Systems
C Compiler	Kuma
C Compiler	Mark Williams Company
C Compiler	Philon
COBOL	Philon
COBOL Compiler	Crupe GmbH
Dev Pac	Hisoft
Fast BASIC Compiler	Philon
Fast C Compiler	Crupe GmbH
Fast PASCAL Compiler	Crupe GmbH
FORTH	Computer One
FORTH Accelerator	The Dragon Group
FORTH 83	Data Becker
FORTRAN Compiler	Crupe GmbH
FORTRAN-77	Philon
FORTRAN-77	Prospero Software
GFA-BASIC Interpreter	Integral Hydraulik
Haba Hippo-C	HABA Systems
Holmes C Duckworth FORTH	Mirage Concepts
ISO PASCAL	Prospero Software
C Soba	KUMA Computers
Lattice C	Kuma Software
Lattice-C	Metacomco
LISP	Metacomco
M-BASIC Compiler	Crupe GmbH
Macro Assembler	Metacomco
Macro Assembler/Editor	Kuma Software
Macro Assembler/Editor	GST Systems
Megamax C Compiler	Megamax
Meta Pascal	Antic Software
Meta Pascal	Metacomco
Micro COBOL Compiler	DRS Software
MODULA-2	TDI Software
p System	TDI Software
PASCAL	Computer One
PASCAL Compiler	Hisoft
PASCAL Compiler	Philon
Personal PASCAL	Optimized Systems Software
Personal PROLOG	Optimized Systems Software
PolyFORTH	Ind. Control Services
Profi Assembler	MDG Software
Profi Mettler	MDG Software
Prolinet 81	DATA Becker
RPG	Philon
ST BASIC	ATARI
ST BASIC Compiler	ATARI
ST Development System	ATARI
ST LOGO	ATARI
ST PASCAL	OCR Software
TBC Compiler	MDG Software
TBC Lint	MDG Software
Turbo C	Borland International
Turbo Pascal	Borland International
UCSD BASIC	TDI Software

POMOĆNI (UTILITIES)

ADS T	ABS Software
BKP Sien	BKP Software
BOS/S Operating System	BOS Software
Business Calculator	ABS Software
P Diskeditor/BKP	C Soft GmbH
BatIC Libraries	ABS Software
Kalender	ATARI
Flash	ATARI
FOCO	Michael Seirel
Control Panel	ATARI
CP/M-I Z Emulator	Soft Design
Directory print	ABS Software
Disk Library	Extended Software Proposal
Disk Monitor	ABS Software
Disk Index	ABS Software
Disk Monitor	B Data
Grabber Disk	ABS Software
Dock	ATARI
Editor-Disk	ABS Software
Fast TOS	ABS Software
File Pool	ABS Software
Find	ATARI
G-Format	G Data
G-Multi	G-Data
G-Protect	G Data
Hippo Dual	Hippopotamus Software
Hippo RAM-Disk	Hippopotamus Software
Hitchhikers Guide to BIOS	ATARI
Holmes & Duckworth Tools	Maggs Concepts
IBM PC Press Development Sys	Helezmar
Joshua	Pitsen Jagesch
K RAM	KUMA Computers
Format	ATARI
M-Copy	Michtron
M-Disk	Michtron
Mi Dupe	Michtron
Micro-C Shell	Beckmeyer Development Tools
Multi	Michtron
Object File	ATARI
Personal Disk I	Optimized Systems Software
Profi Copy	ABS Software
Programmers Calculator	ABS Software
Rhythm	Softechnics
RAM-Disk	G-Data
Resource Construction Set	ATARI
Softspool	Michtron
Source Module (MMGR)	ATARI
Simshade	CDG-EDV-Service
Symbolic Instruction Debugger	ATARI
Trap Grabber	Haase Computersystems
Transit Bookkeeping System	ATARI
Treetime (RVS)	Michtron
VT 52 Emulator	ATARI
Watch	ATARI
YU-105	Xenon
Yutility	Pennon

POSLOVNI

2 Key Accounting System	ATARI
T Key Accounting System	Supplemental
Administrator	Tydler
Appointmaster	Cydler
Bilanz Analyse	Haus der Buchhaltung
Bonplaner	ABS Software
Buchhaltung	Haus der Buchhaltung
Business Administrator	Mindware IS
BS Adria	Bavaria-Soft
BS Faktura	Bavaria-Soft
BS Manager	Bavaria-Soft
P Rolliang	C Soft GmbH
Calendar	Michtron
Cash Manager	Tourgate Software
Cost Trader	Quest International
Best Diary	Paradox Software
Easy Record	Michtron
Electric Calendar	Full Logic Corp
Faktura	Haus der Buchhaltung
Financial Bookbase	Electronic Arts
Fixed Assets	ABS Software
Fibu	Haus der Buchhaltung
Haba Calc	HABA Systems
Haba Checkminder	HABA Systems
Haba Dex	HABA Systems
Haba Custom	HABA Systems
Haba Wills	HABA Systems
Haba Window Calculator	HABA Systems
Haba Window Dialog	HABA Systems
Hippo Personal	Hippopotamus Software
Home Account	Sierra ??
HomePak	Batteries Inc
Integrated 7	Kassen
Inventory Portfolio	BOS Software
Invoicing	BOS Software
Ledger Portfolio System	Batteries Inc
K Calc	Antic Software
K Kasse	KUMA Computers
Faria	Haus der Buchhaltung
K Index	Bavaria-Soft
Lager	Bavaria-Soft
Lohn	Haus der Buchhaltung
Mi Kalender	Michtron
Method	Regler Software
Angelmeter	ABS Software
Nominal Ledger	BKP Software
Nominal Ledgers	Chipsoft
Nominal/General Ledger	Quest International
Payroll	BOS Software
Personal Money Manager	Michtron
Project Leader	Reliable Systems
Project Manager	Planner
Purchase	PPspool
Purchase Ledger	BKP Software
Purchase/Creditor Ledger	Quest International
Rechnungen	Bavaria Soft
Sales	Chipsoft
Sales/Debtors Ledger	Quest International
Sales Invoicing	Quest International
Sales Order Processing	Chelisk Systems
Schedule	Haus der Buchhaltung
Schedule	Rising Star
Side Mist	ABS Software
SM Kunden	SM Software
SM Paper	SM Software
SM Rechnung	SM Software
SP Spreadsheet	SM Software

UCSD PASCAL	TDI Software
UCSD PASCAL 77	TDI Software

GRAFIKA

3 D Interiors	ATARI
Business Graphics	Rising Star
CAD-Design	CPC Erlangen
CAD/CAM	Rising Star
DEGAS	Batteries Inc
Dr Doodle	ATARI
Graphite	MiGraph Inc
Easy Draw	P Graph Inc
EZ-Draw	Progressive Computer Appl
Fleet Street Editor	Mirrorsoft
GEM-DRAW	ATARI
GEM Paint	ATARI
Graphic Artist	Progressive Computer Appl
Hippo Art I	Hippopotamus Software
Hippo Pixel	Hippopotamus Software
Masterpraph	Tydler
Megaloot	Pixel Software
NEO Chrome	ATARI
NEO Images	ATARI
NEO Pixie	ATARI
Object Editor	Rugby Circle
Platine ST	DRIP Becker
Print Shop	Broderbund
Print Shop Graphics Library	Broderbund
Rubber Stamp	Pixel Software
Shape & Icon Editor	Monarch Development Fe
ST Coloring Book	The Dragon Group
Val Draw	Rising Star
Val Paint	Rising Star

KOMUNIKACIJE

ASCII/Parallel Forms	Pinnacle Technology
ASCII Comms	Softworks
BBP	Michtron
Chat	SSI Systems
Flash	AR Technology
Forum	Comnet
Haba Comm	HABA Systems
K Comm	KUMA Computers
Local Area Networks	Software Punch
Mi Term	Michtron
PC/Intercomm	Mark of the Unicorn
S Term	Harald Teschke
ST Talk	Quantum Microsystems
ST Term	Comnet
SuM	PPI Systems
Terminal Emulation Software	Cepsal PIC
Val Mail	Rising Star
VICOM	ATARI
PI IBM Emulator	ABS Software

MUZIČKI

Akkorde im Akkord	Steinberg Waldorf
Good Vibrations	Steinberg Waldorf
Music	ATARI
Music System	Island Logic
Music Writer	Rainbow Music

OBRADA PODATAKA

Adress Fenlent	ABS Software
Autostart	ABS Software
Autoindex	BKP Software
C Adressen/BKP	C Soft GmbH
C Video	P Soft
dB One	Eval Inc
Datapoint	Talent Computer Systems
DBManT PI	DRIP Becker
Datentable	Tydler
DB Master	Stoneware
Grab Adressenverwaltung	Code Software
G-Data	G-Data
H&D Base	Softline
Haba Hbx Phone Book	HABA Systems
Hippo Simple	Hippopotamus Software
Holmes & Duckworth Base	Maggs Concepts
K Data	KUMA Computers
Lassobase	Lassarsoft
Manager	ATARI
Optimum	Dalenbach
Superbase	Precision Software
Video Access I	Quadlar Software
VIP Consultant	VIP Technologies
Tassendata	Autoview Systems

OBRADA TEKSTA

1ST_WORD	GST Systems
ATARI Word	Software Punch
BJ Writer	ATARI
Basswriter	ABS Software
C Text	P Soft GmbH
Express Letter Processor	Maggs Concepts
Final Word	Mark of the Unicorn
GEM Write	ATARI
Haba Business Letters	HABA Systems
Haba Mailmerge	HABA Systems
Haba Writer	HABA Systems
Hippo Spell	Hippopotamus Software
Hippo Word	Hippopotamus Software
K Word	KUMA Computers
Late Write	Port Millions Program
Mailmate	Tydler
Megamax Editor	Megamax
Mighty Mail	Michtron
Mince Editor	Mark of the Unicorn
PaperClip	Batteries Inc
Protext	Mertil&Leshol
Regent Word	Regent Software
Spelling Checker	Regent Software
SpellMate	Tydler
SM Manager	SM Software
SM Text	SM Software
ST Writer	ATARI
Supercorrect	Precision Software
Text Designer PI	DATP Becker
Textomat ST	DATP Becker
VIP Imagelines	VIP Technologies
Word for Word	Bay View Software
Wordmate	Tydler
WordMaster	Paal Inc

Spreadsheet	Rising Star
Stock Control System	Bosky International
Taurus Accounts	Eastern Software
Taurus Invoicing	Eastern Software
Taurus Printers Package	Eastern Software
Textes Stock Control	Energy Software
Timeline	Softechnics
Tele	Rosato
Val Dese 2	Rising Star
VIP Analysis	VIP Technologies
VIP Professional	VIP Technologies
Wasp Accounting Time Records	Whitendale
Wasp Accounts Diary Systems	Whitendale
Wasp Incomplete Records	Whitendale
Wasp Insurance	Whitendale
Wasp Legal Accounting	Whitendale
Wasp Legal Diary System	Whitendale
Wasp Membership Accounting	Whitendale
Wasp Printers Estimating	Whitendale
Wasp Time Recording	Whitendale

OBRAZOVNI

ATARI Planetarium	ATARI
Cemward	Word of God Comp
Hippo Ripper	Hippopotamus Software
Haba Planetarium	Hollise
Homework Helper/Math	Spinnaker
Homework Helper/Words	Spinnaker
Maps and Legends	Antic Software
Master Type	Scarborough
Learn Typing Tutorial	Academy Software
Typewriter PI	Kleni Software

ZABAVA

3 D Chess	Intelligent
A Mind Forever Voyaging	Infocom
Amazon	Spinnaker
Asteroids	Crossbow
Astrologie	DM Software
Bases	Bluechip Software
Battlezone	Freespoon
Biorhythm	Kuhlen&Voges
Bongo	S no Kreppel
Borrowed Time	Activision Inc
Brattacas	Psygnosis
Breakout	ATARI
C Lotto/BKP	P Soft GmbH
Caverns	Paradox Software
Colorspace	Antic Software
Colorspace	Llamasoft
Crimson Crown	Penguin Software
Cut Throats	Infocom
Deadline	Infocom
Delta Patrol	Antic Software
Dragon World	Spinnaker
Eidolon	Epyx
Electro Solitaire	Soft Logic Corp
Enchanter	Infocom
Fahrschul ATS	Spinnaker
Flight Simulator	Michtron
Flight Simulator 2	SubLogic
Flipside	Right on
Forbidden Quest	Priority Software
Gold Runner	Michtron
Goonies	Datasoft
Hacker	Activision Inc
Halley Project	Mindscape
Hex	Mark of the Unicorn
Hippo Backgammon	Hippopotamus Software
Hippo Jokes & Quotes	Hippopotamus Software
Hitchhikers Guide to Galaxy	Infocom
Hyperdrive	English Software
Pyphus	Penguin Software
Impossible Mission	Epyx
Infidel	Infocom
Jet	SubLogic
Jewel	ATARI
Kings Quest I	Sierra On Line
Lands of Havoc	Right on
Lost Kingdom of Zkul	Talent Computer Systems
Major Motion	Michtron
Marble Madness	Electronic Arts
Megaroids	Megamax
Millions ??	Bluechip Software
Millipede	Andromeda
Mindshadow	Activision Inc
Missile Command	Andromeda
Mission Mouse	Antic Software
Mom and Me	Antic Software
Mosaic	Right on
Murray and Me	Antic Software
Mr Phipps 2	Delaware
Nine Princes of Amber	Spinnaker
Pandata	Penguin Software
Perry Mason Mandarin Murder	Spinnaker
Planetfall	Infocom
Polygon	Island Logic
Quiz	Pimpsoft
Quizzes	ABS Software
Seastalker	Infocom
Silent Service	Microprose
Sorcerer	Infocom
Spellbreaker	Infocom
Star Raiders	ATARI
Star Raiders	Crossbow
Star Raiders II	Andromeda
Starcross	Infocom
Starglider	Firebird
Summer Games	Epyx
Summer Games 2	Epyx
Sundog: The Frozen Legacy	FTL Games/Software Heaven
Suspended	Infocom
Time Bandit	Michtron
Transylvania	Penguin Software
Treasure Island	Windham Classics
Typhoon	Bluechip Software
Ultima I	Sierra On-Line
Universe I	Omnitrend
Wall	Talent Computer Systems
Winnie the Pooh	Sierra On Line
Winter Games	Epyx
Wishbringer	Infocom
Witness	Infocom
Wizard of Oz	Windham Classics
Zork	Infocom
Zorro	Datasoft

Računari i obrazovanje

Programirana nastava

škola sa osam programa

Računar u obrazovanju može da se koristi na više načina. Zahvaljujući jeftinim mikroračunarima i sve boljem obrazovnom softveru oko 75% državnih škola u SAD koristi računare za podršku nastave a samo desetak godina ranije ovo moćno nastavno sredstvo koristilo je svega 10% škola. Pri tome je u prvo vreme računar korišćen pravashodno u tzv programiranoj nastavi dok se danas sve više koristi za modeliranje i stimulaciju

Nova vizija ili nova zabluda

Od svoje pojave u SAD pedesetih godina do danas programirana nastava je doživela značajan uspeh i podstakla razvoj i proizvodnju raznih tehničkih sredstava programiranih udžbenika i softvera. Programirana nastava je vrsta nastave u kojoj se sadržaji na poseban način logički strukturiraju i daju učenicima u manjim ranije pripremljenim delovima. Ove celine oni uče samostalno i postupno idući korak po korak sopstvenim ritmom uz stalno proveravanje stepena usvojenosti izloženih sadržaja. Računari kako veliki tajmšering sistemi tako i mikro računari predstavljaju idealno sredstvo za realizaciju programirane nastave

Postoji više vrsta programirane nastave, a programi koji na ovaj način podržavaju učenje sastoje se obično iz četiri dela. Na početku se nalazi *tekstualni materijal* koji objašnjava o čemu se radi. On mora biti jednostavno i jasno napisan. Zatim se zadaju *tekstualna pitanja* koja zahtevaju odgovor učenika. Računar analizira odgovor. Ako je odgovor tačan dobija se potvrda a ako nije ukazuje na probleme na koje treba obratiti pažnju. Uvek su predviđeni i *alternativni tekstovi* na istu temu koje obično pišu razni profesori tako da će verovatno svaki učenik naći pristup problemu koji njemu najviše odgovara. Na kraju se vrši *analiza uspeha* koja pokazuje koliko je tema savladana daje procenat tačnih odgovora i sva druga pokazatelja koji su interesantni za nastavnika.

Tehnološko obrazovanje odraslih

Naš saradnik dr Radomir Mihajlović vanredni profesor na Njujorškom institutu za tehnologiju, radio je u timu koji je za jednu firmu iz Kalifornije projektovao softverski paket namenjen obučavanju za korišćenje novih tehnologija. Projekat je kasnije otkupio IBM i danas je u širokoj upotrebi u svim njegovim fabrikama. Zamolili smo profesora Mihajlovića da nam kaže nešto više o njemu

— Ovaj paket programa trebalo je da omogući da se i ljudi koji nemaju talenta za tehniku obuče za korišćenje i proizvodnju savremene tehnologije. Problem na Zapadu je taj da se tehnologije usavršavaju sve više



Kompjuterom za kompjutere Prof dr Radomir Mihajlović

i više a da je sve manje talentovanih ljudi koji su u stanju da ih prate i koriste. Prema tome u korišćenje nove tehnologije moraju se uključiti i oni sa nešto slabijim intelektualnim kvalitetima. Da bi se jedan čovek netalentovan za matematiku elektroniku i programiranje obučio za produktivno korišćenje savremene tehnologije potrebno mu je posvetiti proporcionalno znatno više pažnje nego nadarenom. On ne može sam da uči — neophodno mu je više tutorijalne interakcije. Međutim ograničen je broj kvalitetnih profesora i oni fizički ne mogu da postignu da stotine sati posvete svakom svom studentu kao što je Aristotel činio sa sinovima bogatih Atinjana. Bogati Grci slali su svoje sinove Aristotelu, on bi šetao s njima kroz park i postavljao bi im pitanja a oni odgovarali. Veštim postavljanjem pitanja navodio bi ih na zaključke tako da su imali utisak da su sami došli do pravih odgovora. Ovaj projekat predstavlja pokušaj da se na računaru simulira Aristotelov metod

Moj zadatak bio je da pripremim programe za učenje o mikroprocesorima. Od hardvera smo imali na raspolaganju računar spregnut sa video diskom čije je važno svojstvo da poseduje vizuelnu informaciju izuzetno velikog kapaciteta. Na ovaj način daje se mnogo efikasnija informacija jer je video informacija mnogo efektnija od govorne. Po mojim proračunima na osnovu nekih formula iz teorije informacija, video informacija je čak 200 000 puta efikasnija nego govorna. Umesto da se zadaje lekcija pa domaći zadatak kao u tradicionalnim metodama obučavanja ovde je predviđen niz sekvenci (sekvenca u programiranoj nastavi predstavlja deo nastavne teme koji je sadržajno i logički povezan, prim autora) kojim se prolazi sopstvenim ritmom u zavisnosti od brzine savlađivanja gradiva

Sa video diskom sa direktnim pristupom spregnutim sa računarom programirana nastava realizovala se na sledeći način. Prvo bi se sa video diska preuzela jedna sekvenca po čijem se odigravanju ekran zamrzavao i postavljalo se pitanje kojim se proveravalo da li je usvojen sadržaj prezentiran sekvencom. Ako je odgovor bio korektan, išlo bi se dalje a ako nije odigrala bi se nova sekvenca koja objašnjava u čemu je bila greška i ponovo bi se očekivao odgovor. Koristili smo dva zvučna i jedan video kanal. Na ovim kanalima su na različit način prezentirani isti sadržaji, tako da se za one kojima jedna priča nije dovoljna mogla odigrati i druga. Tu sam morao da predvidim sve moguće grupe odgovora i sekvencu posle koje bi trebalo da učenik bude u stanju da shvati u čemu je pogrešio i da da ispravan odgovor što mi je moram priznati bilo teže od svega ostalog

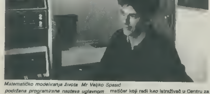
Uz malu pomoć nastavnika

Čini mi se da bez obzira na savršenu tehnologiju i neslućene mogućnosti video diska kao nosioca informacija ipak programirano učenje neće dati prave rezultate bez prisustva nastavnika. Bez obzira na maštovitost tvoraca softvera ne verujem da se mogu predviđati svi mogući odgovori. Mislite li vi da ima smisla učitelja potpuno zameniti računarima, makar oni simulirali i Aristotela?

— Ne nikako. Može se mnogo predvideti ali ne sve razume se na nivou snage računara sa kojima smo mi radili. Međutim ja sam zastupao stanovište da svi ti terminali sa kojih se učilo moraju biti propraćeni jednim moderatorom — tako sam nazvao učitelja koji je trebalo da pomogne u situacijama koje nisu predviđene programom i pruži ljudski kontakt koji za učenika nije ništa manje važan od stručne pomoći u savlađivanju gradiva. Međutim, što se tiče efikasnosti u manipulisanju informacijama čovek je tu ipak znatno ispod mogućnosti računara. Zato smatram da treba vršiti simbiozu mogućnosti računara i nastavnika i tako na najbolji i najracionalniji način realizovati nastavu

Iskustvo profesora Mihajlovića prilog je tezi koja se sve češće čuje da kompjuterski

Dva su viđenja korišćenja računara u obrazovanju — računar kao predmet izučavanja i računar kao nastavno sredstvo. Računari i njihova primena kao predmet proučavanja u nastavi često su bili razmatrani na našim stranicama. O drugoj vrsti njihove primene koja bi trebalo da bude mnogo češća — korišćenja računara kao nastavnog sredstva — još uvek retko ili skoro nikako ne govori. Kako je u našoj zemlji uvođenje računara u učionice još uvek u početnoj fazi, nemamo dovoljno praktičnih iskustava koja bi mogla da se uopšte i ponude univerzalni recept kako napraviti valjan edukativni program i kako ga koristiti u nastavi, ali među našim saradnicima ima i onih koji se profesionalno bave edukativnim softverom, pa bi njihova iskustva mogla biti korisna onima koji žele da na pravi način iskoriste svoje znanje programiranja, kao i nastavnicima i učenicima od kojih se očekuje da koriste gotove edukativne programe.



Matematičko modeliranje života: Mr Veljko Spasić

podržane programirane nastave uglavnom pomaže poboljšanju uspeha lošijih učenika, omogućavajući im da pristignu svoje vršnjake, ali da ima vrlo malo uticaja na razvijanje kreativnih sposobnosti. Posebno ako se zasniva na bihejviorističkoj teoriji koja u središte programirane nastave stavlja odgovor (reakciju) učenika na moguće alternative rešenja i time sputava stvaralačko mišljenje. Uz to se ne sme izgubiti iz vida da se njenom primenom smanjuje mogućnost za razvijanje socijalnih odnosa i kolektivnih veza što, ako školu ne posmatramo samo kao obrazovne već i kao vaspitne ustanove, predstavlja značajnu slabost svake programirane nastave pa i one podržane računarima.

Međutim, razumno i odmereno korišćenje računara kao sredstva za realizaciju programirane nastave omogućava znatno efektivniji rad. Na ovaj način znatno se poboljšava kvalitet organizacije nastavnog procesa. Što je još bitnije, omogućava se individualno usvajanje znanja sopstvenim ritmom, čime se učenici stimulišu za stalno i aktivno učenje. Mogućnosti programirane nastave nisu male. One su korisne naročito u matematici, prirodnim i tehničkim naukama. Primena programirane nastave daje dobre rezultate i u učenju gramatike, sintakse i elementarne pismenosti kao i u mnogim društveno-naučnim predmetima. Kad god su u pitanju saznajne aktivnosti u procesu učenja, programirana nastava može znatno doprineti povećanju efikasnosti, pogotovu ako se u proces učenja uvode problemi koji zahtevaju stvaralačko rešavanje.

Učenje otkrivanjem

Modeliranje i simulacija na računaru u nastavi sve više dobijaju na značaju upravo stoga što kod učenika ne podržavaju mehaničko učenje, već podstiču aktivan misaoni rad. Naš saradnik mr Veljko Spasić, matematičar koji radi kao istraživač u Centru za multidisciplinarne studije Beogradskog univerziteta, duže vreme je radio na Londonskom univerzitetu kao istraživač na naučnoistraživačkim projektima matematičkog modeliranja i simulacija. Paralelno sa tim radio je u centralnom timu CAL (Computer Assisted Learning) u okviru koga je izradio više simulacionih programa od kojih su „Šećer u krvi čoveka" i „Simulacija populacione dinamike" izdati u Velikoj Britaniji (izdavač Longman Publisher, London) i prevedeni u još desetak zemalja. Zamolili smo ga da nam ukratko objasni šta se podrazumeva pod modeliranjem i kako se ono može koristiti u računarski podržanoj nastavi.

— Modeliranje je, u širem smislu reči, postupak zamene jednog objekta drugim koji mu je, u manjoj ili većoj meri, sličan. Matematičko modeliranje je moćno sredstvo u raznim oblastima istraživanja, a koristi se i kao dopuna drugih metoda.

U primeni računara u obrazovanju matematičko modeliranje je najpogodniji način „pakovanja" sistema koji se izučavaju u formi koja omogućava upotrebu računara. Jasno je da su neke oblasti više a neke manje pogodne za ovakav pristup.

Matematički model omogućava pisanje simulacionog programa kojim se prikazuju ponašanje, odnosno osobine modela. Model se posmatra i analizira pomoću računara. Ovakvi simulacioni programi napisani u skladu sa dodatnim zahtevima koje nameće njihovo korišćenje u nastavi, obavezno su praćeni odgovarajućom dokumentacijom. U slučajevima kada je zadatak proučiti složenije, naročito dinamičke sisteme koji imaju komplikovanu strukturu, međuzavisnost sastavnih delova i složeno ponašanje, kao što je to veoma često slučaj u fizici, biologiji, elektronici itd., pristup preko simulacije predstavlja najpogodniji mogući vid prikaza ovih sadržaja. Simulacionim programima mogu se zamenjivati realni eksperimenti i laboratorijske vežbe koje iz određenih razloga ne mogu biti cena, dužina trajanja realnih eksperimenata, nepostojanje odgovarajuće opreme i slično.

Možete li nam opisati neki konkretan simulacioni program?

— Jedan veoma interesantan program engleskog Otvorenog univerziteta (Open University) simulira redove sa čekanjem. Osnovna ideja programa je da omogući lako i fleksibilno izvođenje eksperimenata sa funkcionisanjem uslužnog centra koju ste najpre sami definisali. Na raspolaganju stoje uslužna mesta čiji broj i potrebno prosečno vreme usluživanja zadajete sami. Takođe definišete vreme i trajanje pauze u radu uslužnih mesta, kao i maksimalno dozvoljen broj istovremeno zatvorenih. Korisnici usluga — kupci stižu na slučajan način, a vi određujete njihovu distribuciju. Kada je red prevelik, kupci se vraćaju i ostaju neusluženi, i ovu dužinu reda koja prevazilazi strpljenje kupaca određujete sami. Program, zatim, vrši simulaciju i na ekranu se vide kupci koji dolaze, bivaju usluženi ili odlaze radi gužve. Vidi se red uslužnih mesta i njihovo privremeno zatvaranje radi pauze. Time su rezultati vašeg eksperimenta jasno prikazani. Na kraju se dobija izveštaj koji sadrži podatke o broju usluženih kupaca, broju vraćenih, troškovima rada, profitu, kao i o maksimalnom mogućem profitu. Program izvanredno prikazuje redove sa čekanjem koji se, inače, analiziraju složenim matematičkim aparatom i u vidu eksperimentalne matematike razvija se osećaj za ponašanje sistema masovnog opsluživanja i omogućava slobodno eksperimentisanje.

Kakvu formu treba da imaju edukativni programi? Da li se njihovoj zaštiti posvećuje pažnja kao zaštiti igara i drugih komercijalnih programa?

— Ovi programi nikako ne treba da budu zaštićeni. Jedna od važnih namena je da njihovi korisnici, učenici i nastavnici, mogu da uđu u njihovu strukturu i da ih menjaju. Oni nisu načinjeni da budu zamena već pomoć nastavnicima. Stoga su i propraćeni bogatom i jasnom dokumentacijom. Sem interne dokumentacije koja se bavi specifičnostima samog programa, edukativni programi treba da budu propraćeni posebnom dokumentacijom koja se obraća učenicima.

Kako budućnost školstva kao i budućnost mikroračunara u mnogome zavise upravo od edukativnih programa, uvereni smo da ima smisla posvetiti ovoj temi punu pažnju. Pred našim programerima i nastavnicima je ogroman i odgovoran posao kreiranja programske podrške nastave većine predmeta. Stoga pozivamo na saradnju sve one kojima je problematika korišćenja računara u nastavi bliska, izazovna ili bolna, jer ćemo zajedno svakako brže i bolje uvesti računare u učionice. To je možda pravi put ka — efikasnijoj školi.

Nevenka Spasević

Programski jezici

C (2)

nizak nivo a visoke grane

U slučaju pointera b=a+1, gde a i b pokazuju na podatak tipa long integer (4 bajta širok), adresa b je za 4 veća od adrese a. Aritmetika sa pointer promenljivima se obavlja u odnosu na specificirani tip (veličinu) adresiranih podataka. Korišćenje pointer-podataka podseća bejzik programera na stavove PEEK i POKE (slika 7).

Kad je reč o dodeli memorijskih lokacija C vrlo skromno nudi samo opciju statičke dodele. Tačne adrese moraju biti unapred specificirane i dodeljene u toku vremena prevođenja programa, a ne fleksibilno, zavisno od potrebe u toku izvođenja programa (ALGOL-68).

Indirektno, veštim programiranjem moguće je manipulisati i pojedinačnim bitovima izolovanih memorijskih reči. U nameri da se približi mašinskom kodu što je moguće niže, Denis Riči je predložio više neuobičajenih operatora. Ovi operatori čine C programe pomalo teškim za čitanje, jer uz poznate +, –, * i / operatore, C programer ima na raspolaganju „asemblične" operatore kao što su ++ (inkrement za 1), –– (dekrement za 1), << (pomak ulevo), >> (pomak udesno), & (I, engl. AND), 'I' (ILI, engl. OR), ! (komplement 1, engl. NOT). Da bi bio u stanju da kontroliše ulazno-izlazne jedinice, ekran ili pak, konkretnu memorijsku lokaciju, za sistemskog programera je vrlo važna mogućnost direktnog manipulisanja bitovima. C to u potpunosti podržava i zbog toga ga mnogi u šali nazivaju „asembli jezikom visokog nivoa". Uz povećane mogućnosti sistemskog programiranja ovde se otkriva i tajna brzine izvršenja C programa. Tajna je delimično u eksplicitnosti i prirodnosti „dubinskih" operatora jezika C u odnosu na mašinu.

Jednostavni test program (eng. bechmark) za sabiranje svih celih brojeva po redu od 0 do 15 000 je pedesetak puta brži od istog programa napisanog u bejziku. Sa izuzetkom specijalnih slučajeva, u većini situacija sa kvalitetnim C prevodiocem, razlika u brzini izvršenja istog programa u C i u asembli jeziku je gotovo neprimetna. Za računare sa različitim centralnim procesorskim jedinicama potrebno je naučiti različite asembli jezike. Pri prelasku na novi računar napori i cena ponovnog obučavanja mogu da dostignu neugodan nivo. Ljubitelji programskog jezika C tvrde da je C potrebno naučiti samo jednom i da je to beznačajna avantura u poređenju sa odisejom kroz uzburkane vode mnogobrojnih asembli jezika. Ako ni zbog čega drugog, u svakom slučaju zbog ovoga vredi naučiti C.

Protiv komplikovanja komplikovanog

Svi „paskalični" tj. algolski jezici (engl. Pascal-like/Algol-like) pa i jezik C, svrstavaju se u grupu jezika visokog nivoa. Programski jezik C je malo jezik na visokom nivou. To je ono što je lepo kod bilo kog jezika koji je potrebno naučiti brzo i bez većih problema.



C je, slično algolu, tipiziran i modularan jezik. C program je moguće razbiti na unutrašnje i spoljašnje blokove ili module (sl. 8). Svaki spoljašnji blok ima specijalnu namenu ili funkciju, pa otuda naziv spoljašnje funkcije. Funkcije prihvataju ulazne podatke (parametre) „po vrednosti". Preneseni parametri se stavljaju na raspolaganje potprogramu, uz očuvanje „rezervne kopije" u glavnom programu.

C ne dozvoljava ugnezdivost programa. Funkcije funkcija (deklarisanje nove funkcija u funkciji) ili dalje komplikovanje komplikovanog nije dozvoljeno. Ukratko modularnost C programa uveliko olakšava lociranja moguće greške (engl. dubugging) i dalju dogradnju gotovih programa. Značaj modularnosti je eksponencijalno rastuća funkcija veličine programa. Ko to ne veruje uveriće se brzo ako se prihvati iole većeg programerskog posla.

Uprkos činjenici da C ima manje ključnih ili rezervisanih reči od paskala, može se reći da poseduje daleko veće mogućnosti. C pokazuje svoju nadmoćnost kroz pažljivo uključivanje pogodnih kontrolnih struktura i tipova podataka, dozvoljavajući svojim korisnicima da praktično budu oslobođeni svih ograničenja.

C nudi samo neposredne i jednostavne instrukcija za kontrolu redosleda izvršenja: testove tipa ako-onda-inače, petlje za ponavljanje izvršenja nekih instrukcija, grupisanje instrukcije u blokove i potprograme. Za razliku od jezika ada i modula, kod većine verzija C jezika ne postoje kompleksna kontrolne instrukcije za paralelno izvođenje operacije kao što su to instrukcija za multiprogramiranje i sinhronizaciju ili korutine. (O novim trendovima u sistemskom programiranju i softverskom inženjerstvu uopšte biće reči u jednom od narednih brojeva.) C ne nudi operacije za direktnu odbranu kompozitivnih objekata i kompleksnih struktura podataka tipa lanaca

Pisanje softvera u tehnološkoj situaciji sa svakodnevnim iznenađenjima i novinama, kada se mašine smenjuju preko noći, predstavlja izuzetno složen poduhvat. Pažljivim izborom programskog jezika i ostalih razvojnih programskih pomagala na samom početku projekta, posao se može značajno pojednostaviti. Izbor programskog jezika, međutim, između nije nimalo jednostavan zadatak. Dopisnik „Računara“ iz Njujorka dr Radomir A. Mihajlović piše o kontroverznom jeziku C, koji u višemilionskoj armiji američkih programera poslednjih godina predstavlja pravi hit. Ne bez razloga. Na svetu trenutno nema efikasnijeg, jednostavnijeg i prenosivijeg jezika od jezika C. Da stvar bude lepše, C je dostupan i domaćim programerima, čak i onima koji pišu na „spektrumu“, „amstredu“ ili „komodoru“

karaktera skupova, lista ili matričnih višedimenzionalnih tabela označenih jednim zajedničkim simbolom (PL/I, paskal, itd.).

Operacija na složenim strukturama se mogu „ugraditi" u C prevodilac jednostavnim dodavanjem spoljašnjih rutina postojećoj C biblioteci Kreiranjem sopstvenih funkcionalnih modula programer može udesiti da njegov C bude u stanju da čini čuda Moguće je, praktično, definisati novi, privatni jezik bez ičeg ekstra što programeru ne treba Ovo znači da C jezici različitih korisnika mogu da poseduju različite mogućnosti potpuno prilagođene pojedinačnim potrebama

U svojoj jednostavnosti, primera radi, C nema ugrađene ulezno/izlazne (engl Input-Output ili I/O) programske konstrukcije ili komplikovane programske kontrolere iznenadnih zahteva (engl Interrupt handlers) Za komplikovanija usluga C se obraća spoljašnjoj biblioteci rutina koja mogu pripadati i prevodiocu jezika i operativnom sistemu Ulaz i izlaz podataka u jeziku C kontrolišu lako uočljivi moduli spoljašnjih funkcija **scanf**() ili **printf**()

Upotrebni programi (engl utilities) i programski pretprocesor omogućavaju programeru da izoluje detalje određene datim računarima od glavnog koda C kompajlera U uvodu svakog C programa dotični računar se predstavija nizom specifičnih rutina, sabranim u standardnu ulezno-izlaznu datoteku u zaglavlju **stdio.h** (u našoj verziji C jezika stdioz) Specifičnosti korišćenog sistema su problem programskog pretprocesora i datoteka stdio h Ulaz u C pretprocesor je izvorni C program a izlaz „ispeglana" verzija izvornog programa (slika 8) Ovakav prikaz drastično olakšava ponovno definisanja Ca pri prelasku na novi sistem Vrlo je jasno šta i gde treba menjati u sistemu programa i rutina C prevodioca da bi se C prevodilac za stari računar prilagodio na novi Zaključak je da su minimalnost jezika C i njegova fizička nezavisnost tj labava veza sa spoljašnjom bibliotekom, prava tajna njegove jednostavne prenosivosti između računara najrazličitijih tipova

Zahvaljujući svojim „malim dimenzijama" C je lako prenosiv portablian jezik C program napisan za jedan računar može biti izvršen sa malim ili pak bez ikakvih izmena na drugom računaru sa C prevodiocem u sebi C prevodilac je moguće napisati sa manja od 10 000 linija instrukcija u C jeziku Korišćenjem savremenih programskih pomagala i uz skromnu veličinu C jezika moguće je napisati kompaktan C prevodilac za novi računar i napunih per meseci

```
/* izvorni C program */
#include <stdio.h>

tip funkc_a(..);
{
 ... /* spoljna funkcija */
}

main()
{
 .
  {
   .. /* unutrasnji blok */   ---> C            ---> C
  }                                Preprocesor       Kompajler
 .
 .
 .
}

tip funkc_b(...);
{
 .. /* spoljna funkcija */
}
```

Slika 8.

Uskoro C standard

Široka primena C jezika, slično „juniksu" dovela je do raslojavanja u načinu njegovog korišćenja Interesantno, iako vezan za „juniks" više od 75% korišćenih C kompajlera je instalirano na mašinama bez „juniks" sistema Masovnost i proizvoljnost upotrebe C jezika je neminovno donela na dnevni red pitanja standardizacija — šta bi trebalo a šta ne bi da bude deo osnovne verzija

Juna 1983 godine ANSI (American National Standard institute) je formirao grupu sa zadatkom da pripremi predlog standardne definicija C jezika. Grupa poznata kao X3J11 se redovno sastaje svakog tromesečja. Na sastancima — dostupnim publici, normalno uz cenu ulaznice od 75 koja važi za čitavu godinu — raspravljaju se problemi iz tri generalne oblasti problemi jezika, biblioteka rutina i razvojna sistemske situacija Predsedavajući na ovim sastancima su Lari Rosler (AT&T), P J Plauger (Whitesmiths, Ltd) i Ralf Francr (Konsultant)

Od sledećeg broja

Poliklinika C

Stručni saradnik „Računara“ iz Njujorka prof dr Radomir A Mihajlović pripremio je opširnu školu jezika C — polikliniku u kojoj će „bolesnike“ od bejzika paskala i fortrana pokušati da „izleči“ ovim sve popularnijim programskim jezikom koji de facto postaje svetski standard

Grupa sarađuje sa mnogim zainteresovanim organizacijama iz softverske industrija i poznatim IEEE udruženjem Glavni principi ravnanja grupe su da se nepotrebno ne uveća broj ključnih reči i time ne ugrozi minimalnost Ca, da mu se ne umanji prenosivost i da se ne raskine prirodna veza sa „juniksom" Zvanično usvajanja C standarda očekuje se 1987 godine

I pored neosumnjivih prednosti C nije naročito popularan u Evropi Na drugoj strani okeana, u Sjedinjenim Državama u zemlji sa armijom od nekoliko miliona aktivnih programera jezik C je svakako pravi „hit" Najkritičniji aspekt računarske industrije je relativno nizak nivo produktivnosti u softverskom inženjarstvu Vreme potrebno za projektovanja i realizaciju hardverskog dela računarskog sistema u poređenju sa vremenom potrebnim za razvoj i uspešnu implementaciju softverskog dela je gotovo zanemarljivo

Ma koliko se autori „modernih" programskih jezika treće i četvrte generacije ponosili svojim dostignućima nesavršenost i nedorečenost trenutno raspoloživih softverskih oruđa je evidentna. Enormna dužina programa relativno ograničenih mogućnosti i podebeli softverski priručnici dovoljan su pokazatelj „ubojitosti" modernog softvera. Iz ove perspektiva u poređenju sa zatečenim jezicima C je jezik uvećanih mogućnosti i povećane jednostavnosti upotrebe Kao takav, C definitivno predstavlja krupan a istovremeno i neminovan korak unapred — korak ka savršenim programskim pomagalima budućih generacija programera

Dr Radomir A. Mihajlović

23/nizak nivo a visoke grane

programeri bez zaštite

U Okružnom sudu u Beogradu uskoro će na optuženičku klupu sesti jedan neobičan svat — ovdašnji Gradski elektronski centar — da odgovori na optužbe dva mlada programera da je neovlašćeno preuzeo njihov program za kompjutersku obradu poreskih podataka. Kopiranje i preprodaju softvera u ovoj zemlji niko nikada nije smatrao ni prekršajem ni zločinom, niti je iko ikada zbog toga izašao pred lice pravde. Da li će ovo (za sada) jedinstveno suđenje promeniti naš odnos prema programerima i jednoj novoj, izuzetno važnoj oblasti intelektualnog stvaralaštva?

Zlatne osvajačka godine računara su prošle i nastupilo je vreme surove međusobne borbe za njihov opstanak na tržištu Još pre desetak godina — sa serijskom proizvodnjom jevtinih računara — čovek se od pasivnog svedoka tehnološke revolucije pretvorio u njenog aktera masovnog kupca i korisnika kompjutara U ta konjunktuma vremena neprestano su nicale nove firme novi tipovi računara i sva je imalo prođu

Međutim, publika je u međuvremenu postala znalačka i probirljivija a konkurencija sve veća. Ona je terala proizvođače na neprestana makar i mala poboljšanja i unapređenja koja su ugrađivali u svoje proizvoda Ponekad je — u nedostatku revolucionarnih ideja — to bio samo puki estetski detalj koji će odvući pažnju sa konkurentskog proizvoda Kopiranje programa tehničkih rešenja pa i kompletnih proizvoda sa izmenjenom kozmatikom postali su svakodnevna praksa u proizvodnji računara, koja su se solidarno i prećutno tolerisala

Ali idila je po svemu sudeći završena

Računarski spor u Beogradu

Bilo je pitanje dana kada će se i kod nas — sa razvojem računarske kulture — desiti slični slučajevi I desilo se doduše ne u proizvodnji već u programiranju računara Pred Okružnim sudom u Beogradu našla se tužba dvojice mladih stručnjaka-istraživača protiv Gradskog elektronskog centra Zavoda za statistiku u Beogradu

Rajko Cvetković, inženjer organizacije rada i **Miloš Panić**, ekonomista, smatraju da su oštećeni neovlašćenim korišćenjem njihovog programa za kompjutersku obradu poreskih podataka od strane GEC-a



U nebranom računarskom grožđu Miloš Panić i Rajko Cvetković

Tačnije optužuju ovu gradsku instituciju da je plagirala pozajmila i kopirala deo njihovog programa, prikazujući ga kao svoj!

Kako su se uopšte Cvetković i Panić našli u ovom nebranom računarskom grožđu?

U „inkriminisano" vrame obojica su radili u upravama društvenih prihoda beogradskih opština Zvezdara i Palilula Još 1969 godine Gradski elektronski centar je počeo automatsku obradu podataka na svojim mašinama, da bi devet godina kasnije (1978) svih 16 opština dobilo računare NIXDORF 8820 za takozvano „obuhvatanje" i knjiženje podataka dok je njihova obrada i dalje ostala u nadležnosti GEC-a. Ideja je bila u tome da se elektronski „mozaik" Beograda rastereti bušenja kartica.

Posle dva godine rada na tim mašinama i sagledavanja njihovih mogućnosti, Panić i Cvetković sa još trojicom kolega iz opština Voždovac Rakovica i Stari grad, samoinicijativno su uradili elaborat o modernizaciji poslovanja poreskog knjigovodstva u opštinskim službama Gradski i opštinski oci su januara 1982 usvojili taj elaborat, da bi u sledećoj fazi njegovi autori uradili i programski paket (software) za samostalnu knjigovodstvenu obradu u službama

— Sve smo to radili uz naš redovan posao, tvrdi Cvetković — Sve u svemu, u sledeće dve godine taj je paket „proradio u opštinskim službama ali uz naš veliki angažman Morali smo, naime, da koordiniramo primenu našeg programa, jer su opštine bile deficitarne sa stručnim kadrovima uglavnom su to operativna služba sa bivšim daktilografkinjama i srednjoškolcima

Prema tvrđenju Cvetkovića i Panića, maja 1984 Gradski elektronski centar je pozvao na sastanak predstavnike opštinskih službi gde je odlučeno da počev od '85 prestane primena MOPOK-a (zlosrećnog programa naših sagovornika) Zauzvrat, GEC se obavezao da terminalski poveže opštinska mašine sa svojom velikom mašinom IBM 3031 i naknadno uradi novi, aplikativni softver koji će sve to da podržava

Izgleda da upravo od tog trenutka počinje da se dubi stručni i ljudski jaz između aktera koji će eto završiti na sudu

— Već posle nekoliko meseci ispostavilo se da Gradski elektronski centar nije daleko odmakao u tom poslu nastavlja Cvetković — Krajem godine ponovo su nas pozvali i tražili od nas ponudu za dalji rad MOPOK-a Januara 1985 sročili smo naš zahtev po kome je svaka opština sem dva matične u kojima smo bili zaposleni, treba da nam isplati 80 000 dinara (14 puta 8 jednako 112 starih miliona — prim autora) Posle dva meseca su nas odbili, ne baš učtivo optuživši nas čak da smo — ucenjivači!

Bliži se i završnica ove računarske rašomonijade Gradski elektronski centar je ponovo dobio u zadatak da što pre uradi projekat. Marta prošle godine GEC je opštinama distribuirao jedan deo programa za koji Cvetković i Panić tvrde da je čista kopija MOPOK-a! Na to su ih kako vele upozorile same opštinske uprave prihoda

Tako se na kraju aprila 1985 optužba za povredu autorskih prava koja tvorci programa prethodno nisu ni zaštitili, našla pred nadležnim Okružnim sudom u Beogradu

Šta kaže druga strana?

Jeste — nije kopija

U gradskom elektronskom centru u Tiršovoj 1 nisu nas baš dočekali ozarena lica. Pre bi se reklo — kao nužno zlo. Direktor **Radivoj Gavrić** je na svoju ruku pojačao prijemnu ekipu. Pored njega tu su još i **Nevenko Antunac**, načelnik Gradske inspekcije društvenih prihoda, **Olivera Nikolić** i **Dragoljub Jakovljević**, projektanti u GEC-u. Ništa od one borbenosti, žara i ubedljivosti naših prethodnih sagovornika. Mukotrpno se kristališe tema.

Direktor Gavrić smatra za shodno da podvuče:

— Računari NIXDORF nabavljani su za opštine s ciljem da se vremenom povežu sa IBM mašinom u našem centru. Međutim, od početka su upotrebljavani kao samostalne oprema. To se jednostavno dopalo opštinskim službenicima i tako je zanemaren osnovni cilj.

Ko je za to kriv?

— Inostrani proizvođač nam nije obezbedio softversku podršku za povezivanje mašina. Predstavništvo „Balkanija" je ukinuto a s njim je prestala i ta vrsta obaveza.

Gradski inspektor Antonac praktično je vodio čitav posao koji je na kraju lapao sporan.

— Sa uvođenjem NIXDORF-a školovali su se i kadrovi. Nesumnjivo je da su drugovi koje pominjete dali neka bolja rešenja za uspešnije korišćenja tih računara. Drugim rečima iz novog programa se mogao crpeti veći broj podataka. Zato je i prihvaćen. Međutim, ne radi se o revolucionarnim promenama: sistem poreskog knjigovodstva propisima je strogo regulisan i valja ih se pridržavati.

Antonac tvrdi da je među samim autorima programa došlo do nesuglasice i da su Cvetković i Panić jednostavno odstranili ostale saradnike. — Morao sam čak da smirujem i njih i njihove šefove koji su mi se žalili da redovan posao trpi zbog rada na MOPOK-u.

Nezadovoljni programeri su posebno kivni na Oliveru Nikolić iz GEC-a koja je i glavni potpisnik spornog navodno plagiranog programa. Ona odlučno odbacuje većinu činjenica i optužbi Panića i Cvetkovića.

— Između njihovog i našeg programskog paketa uvek je postojao problem povezivanja. Odluka da odustanu je bila njihova. Po nalogu grada GEC je prihvatio da na svom velikom sistemu uradi deo poslova iz tog paketa. Deo na velikom računaru nije kopija, ali je činjenica da su u drugom delu morala biti zadržane njihova procedura. Da li je to kopija?

Poslednji odgovor na to pitanje daće sudija Vesna Kušić. Sud je već zatražio veštačenje jedne komisije Elektrotehničkog fakulteta koja treba da uporedi dokumentaciju zaraćenih strana.

Staromodni propisi

Da li će propisi pomoći ili odmoći u rešavanju ovog spora, dosad nezabeleženog u našoj računarskoj praksi?

Pravni savetnik u Autorskoj agenciji za Srbiju **Stanka Krstić** kaže da je prodor računara bio brži od zakonskih inovacija. Praktično nijedan važeći propis ne pominje programiranje kompjutera kao autorsko delo. Nema ga u saveznom Zakonu o autorskim pravima iz 1978. a u Zakonu o zaštiti pronalazaka, tehničkih unapređenja i znakova raspoznavanja izričito je naglašeno da se računarski programi ne smatraju pronalaskom.

— Međutim, u praksi mi i računarske programe tretiramo po Zakonu o autorskom pravu — kaže Stanka Krstić. — Oslanjemo se na član 3. u kome stoji da se autorskim delom smatra tvorevina iz oblasti književnosti, nauka, umetnosti i **drugih oblika stvaralaštva**. U skladu s tim vršimo i registraciju programa onih autora koji i na taj način žele da se obezbede od eventualnog plagijata. Pre svega se to odnosi na softverska programa.

Vesna Lazić iz iste agencije tvrdi da se najviše oslanjaju na tumačenja sarajevskog profesora **Pajić Vojislava**, eksperta za autorska prava, koji pominje i računarske programe kao intelektualno autorsko delo. Handikep je u tome što mnogi još smatraju da ovi programi ne služe direktno čoveku poput tradicionalnih autorskih dela, već mašini, dakle tek posredno — čoveku.

U svakom slučaju, ako hoćemo da držimo korak sa tehnološkom revolucijom, programiranje računara zaslužuje bolji tretman od drugih oblasti stvaralaštva. Time bi se ne samo izbegla nemila igra ljudska sujeta, zavisti, emotivnog nadmetanja, krađa i prekrađa ideja, već bi se — što je važnije — ohrabrila generacija mladih stručnjaka programera da se s osloncem na zakonske propise, upusti u nove poduhvate.

Utoliko je i opisani prvi računarski spor u nas ujedno i probni balon društvenog odnosa prema ovom novom obliku tehnička inteligencije.

Mihajlo Kovač

Akcije — Komodor 64

Programi za kertridž

Prve domaće laste

Pokrećući akciju za pisanje programa za kertridž, nismo očekivali preveliki odziv čitalaca. Temu smo, ipak, smatrali dovoljno interesantnom, pa smo bili spremni da gotovo ceo posao preuzmemo na sebe. Na sreću, prevarili smo se. Priloga zaista nije stiglo mnogo, ali po svom kvalitetu i pristupu predstavljaju izuzetnu vrednost. Izgleda da smo ohrabrili jedan prilično slabo poznat segment korisnika računara koji poseduju izvanredno znanje, ali ga koriste isključivo za — svoje potrebe.

Nije nas iznenadilo što se svi prilozi bave grafičkim mogućnostima komodora. Ovo je kao što svi vlasnici znaju Ahilova peta ovog popularnog računara. Zanimljivo je da nigde u radovima nije čak ni pomenuta reč sprajt. Dobre igre se očigledno ne pišu proširenim bejzicima. Naročito nas je iznenadilo što su programeri svoju pažnju usmerili na samo jedan grafički mod i to onej sa najvišom rezolucijom. Multikolor se pojavljuje samo stidljivo — više kao demonstracija mogućnosti da priložene rutine mogu da rade i na taj način. Iznenađenje je svakako prijatno jer pokazuje da autori ovih programa grafiku koriste za ono za šta je i najvrednije — grafičku prezentaciju rezultata neke obrade, predstavljanje funkcija i slično, za šta mogućnost korišćenja više boja nema značaja. Dakle sudeći po vašim reakcijama bili smo potpuno u pravu kada smo tvrdili da nam nisu potrebna velika proširenja tipa Simon's Basic, već kratki i efikasni dodaci koji samo ispravljaju ono što je Komodor propustio.

Posebno bismo hteli da istaknemo dva priloga koje smo dobili od Zoltana Pekića i Dušana Božina.

Zoltan Pekić nas je iznenadio izuzetnim pristupom pri pisanju svojih rutina. Posebno nas je oduševio originalni način razmišljanja — rutina koja na osnovu stanja registara video procesora izračunava adresu video memorije visoke rezolucije. Dakle, gde god se nalazila, na koji god način da je pozvana, pozicija slika visoke rezolucije će biti tačno izračunata i zatim kao parametar biti preneta u PLOT, LINE i slične rutine. Pošto ćemo u konačnom programu obavezno imati i naredbu za izbor režima rada video procesora, tačno ce biti definisano upotrebom nekoliko novih sistemskih varijabli koji je režim izabran i gde se video memorija nalazi. Zato će na žalost ova rutina biti suvišna. Ostale rutine PLOT i LINE takođe su odlične. To se posebno odnosi na PLOT koja je efekino rešena upotrebom tablica rasporeda bitova čime je na račun nešto veće potrošnje memorija dosta dobijeno na brzini. Za tako često pozivanu rutinu ovo je od primarnog značaja, pa ćemo u potpunosti prihvatiti njegovo rešenje.

Druga Pekića smo zamolili da svoj prilog uobliči prema manjim sugestijama koje smo dali pa njegov rad u celosti očekujemo u nekim od sledećih brojeva „Računara".

Dušan Božin je poslao svoje rešenje PLOT rutine. Da je kojim slučajem Komodor njemu poverio pisanje programa za ROM-ove C64, sigurno je da bi potrebno područje za sistemske varijable bilo upola manje! Dušan na najbolji način ilustruje kako se veštim korišćenjem registara i steka pišu brze rutine bez čestog obraćanja memoriji. Njegov prilog, ipak, nešto odstupa od koncepcije koju smo na početku usvojili. Osnovno je da sva rutine koje pripadaju operativnom sistemu budu potpuno nezavisne od toga kako će biti primenjene u proširenjima bejzika. Ulazni parametri (konkretno X Y koordinate i modalitet postavljanja tačke) treba da se prenose u rutinu na način primeren operativnom sistemu — dakle putem registara ili memorijskih lokacija a ne na bejzik način uzimanjem parametara iz bejzik linije. Time PLOT postaje dostupan i programima u mašinskom jeziku, što je u drugom slučaju nemoguće. Dušan kaže da ima još i DRAW i TEXT rutine koje želi da doradi pa će, ako nam se dopadnu ove koje je poslao, poslati i njih. Pozivamo ga da to obavezno učini uz molbu da uvaži generalnu primedbu koju smo dali.

Dušan je postavio i pitanje kako će izgledati nove bejzik naredbe koje budemo uveli — da li će to biti pune reči ili samo slova uz neki zajednički prefiks? To ni nama još uvek nije do kraja jasno. Ako to budu pune reči, biće neophodno menjati rutine za tokenizaciju i listanje programa, što može odneti dosta memorije. Ako pak, budu oblici tipa "P i slični, imaćemo više mesta za korisnije stvari ali i slabiju razumljivost programa. Dakle na vama je da donesete odluku.

Sve one koji nameravaju da se uključe u akciju molimo da, ako su programi iole duži, prilože i kasetu sa izvornim programom kako bismo ubrzali postupak oko provere priloga i njihovog objavljivanja. Oni koji nemaju štampač ne treba da brinu — kaseta će biti dovoljna a listing ćemo napraviti mi.

Zoran Životić



Peek & poke show

Mikrotragedija

Dogodila se jedna tužna stvar Hiljada novinarskih računardžija i računarskih novinara na ovom belom svetu neće više moći da kinje i maltretiraju svoju omiljenu žrtvenu figuru Ser Klajv je napustio računare! Nije umro Jednostavno je prodao svoju firmu nekom drugom i otišao da se bavi nekim drugim stvarima (telefonima i slićnim cakama) To je korak koji se može nazvati izdajstvom ali i kada se setimo kako mu je išlo u poslednjih godinu-dve manom zdravog razuma

Onaj drugi tačnije Onaj Drugi koga smo pomenuli kao kupca nije niko drugi do sjajni i nepredvidivi Alan Šugar-Šećerko uspešni i ponosni direktor Amstrada Tako je obavljena direktna primopredaja — najsmotaniji najlukavijem

Vlasnici malih vrućih crnih kutija i, uopšte zaljubljenici u spektrum i slična mašine ne moraju da strahuju Šugar je izjavio da neće biti drastičnijih promena u kompaniji Osim što je sad mnogo verovatnije da će QL a možda čak i spektrum" dobiti nekakav disk drajv recimo od 3 inča Toliko o tužnim vestima A sada i jedna dobra — Ser Klajv je napustio računare ali mu je ostala plemićka titula

Naš zajednički računar

Čitaocima koji pažljivije prate dnevnu štampu sigurno nije promakio da je za potrebe Skupštine Jugoslavije nabavljen jedan oveći kompjutar sa mnoštvom monitora štampača disk jedinica i sličnih potrepština Peek & poke show je oduševljen tim događajem i smatra da je ovo prvi put da je stvarnost pretekla našu Čip Pobodi Agency Ima li ubedljivijeg načina da se pokaže da je zaista potrebno masovno uvoditi kompjutere u našu stvarnost od praktičnog primera? A šta je praktičnije nego postaviti jedan lep i moćan računar tako da ga vidi mnoštvo naših vodećih i rukovodećih ljudi A gde je to? U Skupštini Peek & poke show je uveren da je dalja taktika već razrađena i da će se u dogledno vreme mereno danima slični (malo manji i malo slabiji) računari, kao logičan korak pojaviti i u skupštinama republika i pokrajina nešto kasnije i u opštinama a jednog dana ko zna možda čak i u mesnim zajednicama Nakon toga će nadamo se na red doći i škola



Top lista „Crni biseri"

SUBGO, KEPO, TOGO, RINTP

Naučite kompjuter da govori šatrovački

U najnovijem broju jedne domaće revije (između ostalog !) za kompjutere pojavio se senzacionalan program dat u vidu listinga, kojim ćete naučiti (naterati?) svoj kompjuter da govori šatrovački Da, dobro ste pročitali šatrovački odnosno slang Ovaj pravi biser domaćeg programerskog umeća, koji odražava trenutni (visok) nivo istog trebalo bi da poseduje svaki pravi korisnik kompjutera

Na žalost program je dat samo za komodor 64' pa će korisnici ostalih tipova računara biti vrlo uskraćeni Ali poznavajući agilnost i entuzijazam domaćih programera ne sumnjamo da će se uskoro pojaviti prepevi i za ostala računare

Predlažemo da ovaj program automatski skoči na prvo mesto naše nove top liste ‚Crni biseri', koja trenutno izgleda ovako

1. *Šatrovački za C-64*
2. *Proračun malih trafoa*
3. *Memo*
4. *Proračun aktivnih filterskih skretnica*
5. *Štrumpfmatika*

Pozivamo čitaoce da nam šalju svoja predloge za ovu listu Broj predloga nije ograničen a najvrednijima za nagradu šaljemo knjigu Prva šljapkanja po čejziku

Veliži karihal

Čip Pobodi Agency

Budući Oskar

Od svojih doušnika u svetu filma i gubljenja vremena saznajemo da poznati režiser Džon Karpenter poznat po svojim horor remek-delima, priprema novi film po scenariju Andy MC Colucke pod radnim naslovom , Bio sam YU pirat i preživeo sam Prema ovim istim doušnicima glavnu ulogu bi trebalo da igra Silvester Stalone Iako je on kao uslov za igranje zahtevao da se dopiše nekoliko scena tuče i razmena programa

Jezici u sosu

Posle fortrana prologa čejzika i kobola kao i drugih jezika koji su do sada korišćeni za eksperimente na srednjoškolcima sve se glasnije čuju zahtevi da se uvede jedan od novih User-friendly jezika za obuku mladih Jezik se zove Boulderdash III i čim čujemo njegove osnovne karakteristike javićemo vam

Kvantni skok nadole

Nobelova fondacija je odlučila da proširi spektar nagrada koja svake godine dodeljuje Od ove godine postojaće i Nobelova nagrada za stvaranje električnih tricikala i mrtvih tastatura Tako će najbolji izumitelji sveta moći da budu zaista nagrađeni za svoj izuzetan trud

Posle višegodišnjih pregovaranja firma Drp elektronics izbacuje na tržište napravu koja bi trebalo da omogući da ORIC emulira IBMa PC Naprava je proizvedena po zahtevima jugoslovenskog tržišta i težiće oko 86 kilograma Cena će joj biti između 70063 dolara i 857634547580 dinara Očekuje se da će biti široko prihvaćena

Primer 6:
Generator slučajnih brojeva

```
3600             50   ORG 63600
3600             60   ENT ?
35F0             70 SEED EQU 63500
3600             80 RNDGEN
3600 2AF035      90   LD HL (SEED)
3603 7C         100   LD A H
3604 E6E8       110   AND 4E8
3606 57         120   LD D A
3607 7C         130   LD A H
3608 E601       140   AND 4B
360A 07         150   RL A
360B 07         160   RL A
360C 07         170   RL A
360D 07         180   RL A
360E AA         190   XOR D
360F 5F         200   LD E A
3610 7D         210   LD A D
3611 E606       220   AND 48
36[illegible] 0F          230   RR A
36[illegible] 0F          240   RRCA
3615 AF         250   [illegible] D A
[illegible]         260   [illegible] A L
3[illegible]        270   ADD 6
36[illegible] AA         280   [illegible] D
36[illegible]            290   [illegible] A
3[illegible] AB         300   [illegible] E
[illegible]         310   RLA
[illegible]         320   R L
36[illegible]        330   [illegible]
[illegible] 22F035   340   LD (SEED) [illegible]
36[illegible] 9        350   RET
3[illegible]        360   >
```

Kada smo već kod igara, navedimo i jedan generator slučajnih brojeva. Program sa naše slike se poziva jednostavnim CALL RNDGEN i vraća u HL slučajan broj između 0 i &FFFF. Princip rada je prilično jednostavan (zato generator nije naročito dobar, ali je bar brz) pa ćemo vam ostaviti da ga sami otkrijete. Ako nećete previše da se zamarate, zapamtite jedino da pre *prvog* poziva RNDGEN treba u ćelije SEED i SEED+1 upisati bilo koji šesnaestobitni broj različit od nule.

Dejan Ristanović

Z 80 mašinac za početnike

Ovaj umetak, kao što mu i naslov govori, treba da upozna apsolutnog početnika sa programiranjem na mašinskom jeziku mikroprocesora Z80. Takav apsolutni početnik će se, čitajući ove redove, možda zapitati šta je to mikroprocesor ili odakle oznaka Z80. Odgovore na ta i druga pitanja o značenjima osnovnih pojmova koje ćemo koristiti kroz čitav umetak treba da pruži ovo uvodno poglavlje.

Mikroprocesor je centralni deo svih kućnih računara i različite druge opreme koju nazivamo inteligentnom. Ovaj termin označava čip koji može da se programira za procesiranje, tj. obradu nekakvih podataka. Ono *mikro* u njegovom imenu potiče od činjenice da se čitava komplikovana elektronika nalazi u jednom jedinom integrisanom kolu koje ćete, ako se odvažite da skinete poklopac vašeg kompjutera, obično prepoznati po tome što ima najveći broj nožica, tzv. pinova. Kao i kod većine reči koje počinju sa *mikro*, postojala su i postoje njihove verzije bez tog prefiksa: *procesorom* ili *procesorskom pločom* nazivamo deo nekog velikog kompjuterskog sistema koji obrađuje informacije.

Moderni kućni računari u sebi sadrže više čipova koji bi se mogli svrstati pod našu definiciju mikroprocesora: ukoliko, na primer, vaš računar povežete sa diskom, kontrolu ovoga uređaja će obavljati disk kontroler, tj. specijalizovani mikroprocesor. Ako pročitate da vaš kompjuter ima video kontroler, znaćete da se radi o još jednom mikroprocesoru koji se brine za generisanje slike. Sličnih primera ima još dosta. Mi se, međutim, nećemo baviti specijalizovanim kontrolerima, već mikroprocesorom Z80 koji se nalazi „u centru" dobrog dela danas popularnih modela kućnih računara. Centralni mikroprocesor je zadužen za izvršavanje mašinskih programa koje korisnik startuje (dok, na primer, izvršavate bejzik program, traje izvršavanje bejzik interpretatora koji je upisan u ROM vašeg kompjutera) i za koordinaciju rada ostalih specijalizovanih kontrolera ugrađenih u računar. Korisnik obično nema nikakvih mogućnosti (a ni potrebe) da komunicira sa bilo kojim hardverskim modulom svoga računara osim posredstvom centralnog mikroprocesora.

Z80 jasno nije pretplaćen da uvek budu glavni: ako posetite neki računski centar, saznaćete da se za komunikaciju centralnog procesora (bez onoga *mikro*) sa terminalima brine jedan mikroprocesor, dok drugi neprekidno komunicira sa paketom diskova. U poslednje vreme se mogu pronaći i kućni računari iz srednje klase koji poseduju po dva ili tri različita mikroprocesora, od kojih se jedan brine za ulaz i izlaz podataka a drugi za opsluživanje zahteva korisnika. Za nas je, međutim, sasvim dovoljna pretpostavka da naš kompjuter ima samo jedan mikroprocesor.

Što se oznake Z80 tiče, u njoj ne treba tražiti neki poseban smisao. Z dolazi od imena firme *(Zilog)* koja proizvodi ovaj čip. Ono *80* verovatno treba da podseti da su Z80 proizveli bivši Intelovi konstruktori koji su neposredno pre toga dizajnirali i 8080 mikroprocesor sa kojim Z80 ima dosta sličnosti i od koga je značajno moćniji. Moguće je da je u vaš računar ugrađen

mikroprocesor Z80A Z80B ili neki sličan To ne treba da vas zabrinjava radi se o nekoj od verzija Z80 koja se od osnovne verzija razlikuje samo u nekim hardverskim detaljima mašinski jezik je identičan onome koji mi obrađujemo

Bit, bajt, reč

Čitajući kompjuterske časopise verovatno ste saznali da je Z80 **osmobitni** mikroprocesor a da postoje i **šesnaestobitni** kao i nekakvi za koje nije baš jasno kakvi su (jedan takav je ugrađen u kontraverzni Sinclairov QL) Izgleda smešno da nekome ko već izvesno vreme poseduje računar i ko bar jednom nedeljno se prijateljima raspravlja o tome koji model kompjutera ima više kilobajta memorije objašnjavamo šta je bit i šta je bajt Iskustvo nam međutim pokazuje da se ovakvi razgovori zasnivaju na relativnom poređenju podataka jasno nam je da je računar koji košta 400 australijskih dolara duplo jeftiniji od računara koji košta 800 australijskih dolara ali nam to ne pomaže da znamo koliko *stvarno* košta svaki od ovih računara da bismo to saznali treba da znamo i kurs australijskog dolara Zato ćemo uzeti slobodu da jednom za svagda objasnimo pojmove **bit, bajt i kilobajt**.

Bit je najmanja jedinica za količinu informacija. Ako vas neko pita *da li si kupio Računare 15* možete da odgovorite sa **da** ili sa **ne** (ako odgovorite sa *ne* a ipak čitate ovaj tekst pa kod nas se i onako smatra da krađa knjiga i časopisa nije krađa) Uz malo dobre volje odgovoru **da** možete da pridružite broj 1 a odgoru **ne** broj 0 Za odgovor na pitanje je dakle dovoljan samo jedan bit informacija.

Postoje naravno i pitanja na koja se ne može dati jednobitni odgovor Na pitanja *koliko koštaju Računari 15* ne možete da odgovorite sa **da** ili **ne**, biće potrebno da navedete neku brojku Ta brojka se videćemo u sledećem poglavlju može pretvoriti u drugu brojku koja se sastoji od jedinica i nula pa ćete tako možda reći da časopis koji čitate košta 100101100 dinara Pošto ovaj broj ima 9 binarnih cifara (bit = binary digit ili binarna cifra) odgovor na pitanje o ceni ove knjige nosi 9 bita informacija.

Bit je kao što vidimo vrlo mala količina informacija kao što nikome neće pasti na pamet da izražava rastojanja od Zemlje do Sunca u milimetrima tako se i kapacitet memorija nekog kompjutera nikada ne izražava u bitovima Obzirom da su stepeni dvojke (2*1 2*2 2*3) vrlo značajni za rad sa binarnim brojevima uobičajeno je da se 2*3 = 8 bitova nazivaju bajt Broj koji smo ranije naveli kao cenu našeg časopisa bismo sada napisali u obliku *1 00101100* i rekli da on nosi jedan bajt i još jedan bit informacija (primetimo da je to isto što i 1*8 + 1 = 9 bita) Nije međutim uobičajeno izražavati količinu informacija na ovako "razlomljen" način pa ćemo reći da su za izražavanje cene Računara potrebna dva bajta informacija naših devet bitova ćemo da bismo se uklopili u ovu definiciju dopuniti vodećim nulama pa ćemo tako reći da Računari 15 koštaju *00000001 00101100* dinara Uvođenjem vodećih nula smo uzgred ostavili dosta prostora za buduća poskupljenja

Osam bita ili jedan bajt je najmanja količine informacija kojoj mikroprocesor može da pristupi To znači da će na primer naredba LOAD A (20000) dovesti u akumulator (jedan od registara procesora nije bitno ako ovaj termin još ne razumete) sadržaj memorijske ćelije čija je adresa 20000 i to svih osam bita koje sadrži ta ćelija Ukoliko nam je potrebno da pojedinačno menjamo neke od tih bitova moraćemo da dovedemo čitavu ćeliju u akumulator modifikujemo je a zatim je celu vratimo u memoriju

Videli smo da mikroprocesor može (i mora) da u memoriju odjednom dovede 8 bita informacija ili jedan bajt Zbog ove karakteristika mikroprocesor Z80 nazivamo **osmobitnim** Danas pravo građanstva sve više stiču **šesnaestobitni** mikroprocesori koji dakle u jednom pristupanju memoriji mogu (i obično moraju) da prenesu 16 bita podataka Na tržište se stidljivo probijaju i skupi **tridesetdvobitni** mikroprocesori dok neki veliki sistemi osim tridesetdvobitnih koriste i **šezdesetčetvorobitne** procesore U principu je (mikro)procesor koji može da obrađuje podatke koji se sastoje od većeg broja bitova brži i sposobniji za kontrolu veće operativne memorije ali je zato takav (mikro)procesor kao i ostali čipovi koji ga prate daleko skuplji

Kada već govorimo o šesnaestobitnim procesorima moramo da pomenemo još jednu definiciju bajta Bajt smo naime definisali kao osam bita jer naš mikroprocesor može odjednom da ih pročita baš toliko Ako radimo sa šesnaestobitnim procesorom imalo bi smisla definisati bajt kao 16 bita što su neki autori i radili Bolje je međutim da za šesnaest bita usvojimo termin **reč (word).** Kako su se procesori razvijali njihovi konstruktori su morali da usvoje i termina **double word** (32 bita) a onda i **long word** (64 bita) sledeća veća jedinica za količinu informacija **double long word** (128 bita) još nije zaživela u praksi U terminologiji mikroprocesora Z80 **bajtovi** i **reči** će nam biti sasvim dovoljni

U literaturi posebno domaćoj srećemo još jednu skrivenu definiciju reči bajt često ćemo pročitati nešto poput ova naredba stavlja broj iz akumulatora u pedeset hiljaditi bajt memorije Znajući da je bajt jedinica za količinu informacija gornja rečenica je sasvim besmislena da li ste nekada čuli da je neko sipao flašu vode u deveti litar posude? Kada bi međutim pomenuta posuda bila na neki podeljena u šesnaest komora a u flaši bio tačno litar vode rečenica sipao sam flašu vode u deveti litar posude bi mogla da se protumači kao sipao sam jedan litar vode iz flaše u komoru posude čiji je redni broj 8 vidimo da smo na prvi način uštedeli 15 reči Pošto je memorija računara podeljena na ćelije od kojih u svaku staje po osam bita rečenica ova naredba stavlja broj iz akumulatora u pedeset hiljaditi bajt memorije bi mogla da se prevede kao ova

```
3E3F C5       350       [illegible] DE
3E40 E5       360       PUSH HL
3E41 CD[illegible] 370  CALL [illegible]
3E44 E1       380       POP HL
3E45 C1       390       POP BC
3E46 EB       400       [illegible]
3E47 C9       410       RET
3E48          420 [illegible]
3E48          430       ; [illegible]
3E48          440       ; [illegible]
3E48 C5       450       PUSH BC
3E49 D5       460       PUSH DE
3E4A E5       470       PUSH HL
3E4B CD[illegible] 480  CALL [illegible]
3E4E E1       490       POP HL
3E4F D1       500       POP DE
3E50 C1       510       POP BC
3E51 C9       520       RET
3E52          530 ULAZ
3E52          540       ; OČITAVANJE TASTATURE
3E52 E5       550       PUSH HL
3E53 C5       560       PUSH BC
3E54 D5       570       PUSH DE
3E55 CD[illegible] 580  CALL [illegible]
3E58 D1       590       POP DE
3E59 C1       600       POP BC
3E5A E1       610       POP HL
3E5B C9       620       RET
3E5C          630       ; TEKSTOVI PORUKA
3E5C [illegible] 640 [illegible] TEXT "POGODIO SAM (OPET)!"
[illegible]   650       WORD [illegible]
[illegible]   660 [illegible] TEXT "ZAMISLI BROJ IZMEDJU 1 I 1023"
[illegible]   670       BYTE [illegible]
[illegible]   680 TEXT "A JA CU GA POGODITI!"
[illegible]   690       WORD [illegible]
[illegible]   700 TEXT "POSLE SVAKOG MOG POKUSAJA"
[illegible]   710       BYTE [illegible]
[illegible]   720 TEXT "PRITISNI "
[illegible]   730       WORD [illegible]
[illegible]   740 TEXT "  M AKO JE TVOJ BROJ MANJI;"
[illegible]   750       BYTE [illegible]
[illegible]   760 TEXT "  V AKO JE TVOJ BROJ VECI;"
[illegible]   770       BYTE [illegible]
[illegible]   780 TEXT " - P AKO SAM POGODIO."
[illegible]   790       WORD [illegible]
[illegible]   800       BYTE [illegible]
371C          810   >
```

Možda ste se odlučili za upoznavanje mašinskog jezika da biste zatim pisali igre (i prodavali ih Englezima)? Evo jedne sasvim jednostavne igre primerene znanju koje smo do sada stekli High Low

Zamislite broj između 0 i 1023 i startujte naš program Računar će ispisati neki broj a zatim ćete pritisnuti V ako je broj koji ste zamislili veći M ako je manji a P ako je računar pogodio Videćete da će svaki broj koji zamislite biti pogođen u najviše 10 pokušaja

Igra je sasvim nezavisna od bejzika što znači da ulaz i izlaz podataka obavlja sopstvenim potprogramima ISPIS ISPIS_HL i ULAZ Ovi potprogrami sa svoje strane pozivaju rutine iz ROM a čije adrese možete da saznate konsultovanjem knjiga tipa The Complete Spectrum ROM Disassembly ili naših ranijih umetaka Moraćete dakle da promenite adrese navedene u naredbama 50 60 i 70

Primetite da smo ulaz i izlaz podataka odvojili u posebne potprograme vodeći računa o činjenici da potprogrami iz ROM-a nekih računara ne čuvaju sadržaje opštih registara Preporučljivo je slediti ovakvu metodologiju rada ako naredbe zavisne od računara izdvojimo u potprograma i dobro dokumentujemo naša će se remek dela lako prilagoditi drugim mašinama

Što se algoritma na koma počiva program High Low tiče nećemo mu obraćati mnogo pažnje radi se o običnom binarnom pretraživanju Ukoliko vam je termin „binarno pretraživanje" stran pogledajte mali bejzik program koji je potpuno ekvivalentan bitnom delu našeg mašinca

```
10 PRINT "Zam s" broj izmedju 1 i 1023 "
20 PRINT
30 LOW=0
40 [illegible]
50 [illegible]
60 PRINT "Da li je to broj " BROJ " (V M P)?"
70 INPUT "" X$
80 IF X$="V" THEN LOW=BROJ GOTO 50
90 IF X$="M" THEN [illegible] GOTO 50
100 IF X$<>"P" THEN GOTO 80
110 PRINT "Pogodio sam (opet)!"
120 PRINT
130 GO TO 10
```

```
3610 CD0836 260    CALL NIBL ; ISPISI GA
3622 F1     270    POP AF
361E E60F   280    AND 0FH   ; IZDVAJA NIZI NIBL
361A CD0836 290    CALL NIBL ; I ISPISI GA
361D C9     300    RET
361E        310 NIBL
361E        320    ; UZIMA ASCII VREDNOSTI
361E        330    ; BROJA IZ A U CELIJU 02
3620        340    ; KOJU POKAZUJE DE
361E FE0A   350    CP 10
3620 FA3536 360    JP M,CIFRA
3623        370    ; RADI SE O VREDNOSTI
3623        380    ; KOJU TREBA DA PORTAZU
3623        390    ; SLOVO A-F
3623 C607   400    ADD 7
3625        410 CIFRA
3625 C630   420    ADD "0"
3627 12     430    LD (DE),A
3628 13     440    INC DE
3629 C9     450    RET
362A        460    >
```

```
3600        10    ORG 03600
3600        20    OPT 7
3600        30 BLANC
3600 7C     40    LD A,H
3601 FE00   50    CP 0
3603 2803   60    JR Z,OVE   ; DA LI JE VECE OD 255?
3605 CD0036 100   CALL BAJT  ; ISPISI VISI BAJT
3608        110 OVE
3608 7D     120   LD A,L
3609 CD0036 130   CALL BAJT  ; ISPISI LOW BAJT
360C C9     140   RET
360D        160 BAJT
360D        170   ; UPISUJE ASCII VREDNOSTI
360D        180   ; BAJTA IZ A U CELIJU 02
360D        190   ; KOJU POKAZUJE DE
360D F5     200   PUSH AF
360E E6F0   210   AND 0F0
3610 1F     220   RRCA
3611 0F     230   RRCA
3612 0F     240   RRCA
3613 0F     250   RRCA      ; IZDVOJI VISI NIBL
```

operaciju izvršiti još jednom izazivajući povratak u glavni program. Da smo umesto CALL NIBL RET napisali JP NIBL (ili još bolje, JR NIBL štedeći dva bajta), naredba RET bi bila nepotrebna program bi brže radio i stek bi se manje opterećivao (znate li zašto?). U većini slučaja međutim nije vredno uštedati par bajta i učiniti program nečitljivim što je program slabije strukturiran i što više „prljavih trikova" koristi to će u njemu biti više grešaka te se greške u krajnjoj instanci lupaju o glavu vama ili nekom nesrećniku koji će docnije pokušati da modifikuje ili bar razume vaš program. Obzirom na činjenicu da je asembler po svojoj prirodi nepregledan treba se truditi da se programi što propisnije pišu i što bolje komentarišu. Od ovoga možemo da odstupimo samo u (danas veoma retkim) situacijama u kojima su dragoceni bajtovi i mikrosekunde procesorskog rada.

Primer 5: Jedna jednostavna igra

```
3600        10    ORG 03600
3600        30    OPT 7
3600        20    ; ADRESE RUTINA ZA ROM-2:
0000        50 POMAKNR EQU 0A0
0873        60 POMALINR EQU 0873
0C05        70 KONTAKTI EQU 0C05
3600 217036 80    LD HL UPUT ; ISPISIVANJE UPUTSTVA
3603        90 IGRI
3603        100   ; ISLIKUJE ZNAK POCEVSI OD
3603        110   ; MEM. CELIJE 02 KOJE HL
3603        120   ; POKAZUJE DO NEVOG 0 BAJTA
3603 7E     130   LD A,(HL)
3604 A7     140   AND A      ; DA LI JE NULA?
3605 2808   150   JR Z,DALJE
3607 CD3E36 160   CALL ISPIS
360A 23     170   INC HL
360B 18F0   180   JR IGRI
360D        200 DALJE      ; POCETAK PARTIJE
360D 010000 210   LD BC 0    ; LOW = 0
3610 210004 220   LD HL 1024 ; HIGH = 1024
3613        230 PROBAJ
3613 50     240   LD D,B
3614 69     250   LD L,C
3615 09     260   ADD HL,BC  ; HL = HIGH+LOW
3616 CB3C   270   SRL H
3618 CB1D   280   RR L       ; HL = HL/2
361A CD0836 290   CALL ISPISHL
361D 3E00   300   LD A 0D0   ; NOVI RED
361F CD3E36 310   CALL ISPIS
3622        320 UNOS
3622 CD3236 330   CALL GLAS
3625 FE4D   340   CP "M"
3627 2820   350   JR Z,MANJE
3629 FE56   360   CP "V"
362B 2800   370   JR Z,VECE
362D FE50   380   CP "P"
362F 20F1   390   JR NZ UNOS ; NEISPRAVNO SLOVO
3631        400   ; BROJ JE POGODJEN!
3631 21C036 410   LD HL ZNAK
3634 18C0   420   JR PISI
3636        430 VECE
3636 50     440   LD B,H     ; LOW = HL
3637 4D     450   LD C,L
3638 18D0   460   JR PROBAJ
363A        470 MANJE
363A 54     480   LD D,H     ; HIGH = HL
363B 5D     490   LD E,L
363C 18D5   500   JR PROBAJ
363E        510 ISPIS
363E        520   ; ISPISIVANJE ZNAKA IZ A
```

naredba prenosi osmobitni broj iz akumulatora u memorijsku ćeliju čija je adresa (redni broj) 50 000. Kao i ranije izražavali smo se manje precizno da bismo uštedeli nekoliko reči.

Memorijski registri i memorijske ćelije

Memorija se sastoji od ćelija u koje staje po osam bita i koja su numerisane po svojim rednim brojevima. Pretpostavimo da našem računaru treba da naredimo da u ćeliju čiji je broj 100 upiše zbir sadržaja memorijskih ćelija čije su adrese 101 i 102. Bilo bi logično da se za takvu naredbu upotrebi oznaka

ADD (101) (102) (100)

ADD je skraćenica za naredbu „saberi" i prate ja tri adrese: adresa memorijske ćelije u kojoj je prvi sabirak, adresa drugog sabirka i adresa ćelije u koju treba staviti rezultat (smisao zagrada ćemo razumeti docnije). Računar koji bi mogao da izvrši ovakvu naredbu bi se nazivao *troadresni računar*. Pokazalo se, međutim, da bi navođenje velikog broja adresa iza svake instrukcije predstavljalo neracionalan utrošak memorije, pa su se pojavili dvoadresni i jednoadresni računari. Jasno je da jedna adresa nije dovoljna da potpuno odredi sabiranje koje smo upravo naznačili. Zato će program ekvivalentan gornjoj naredbi na jednoadresnom računaru glasiti

LOAD A, (101)	Prenesi u akumulator sadržaj ćelije 101
ADD A, (102)	Dodaj na sadržaj akumulatora sadržaj ćelije 102, rezultat sabiranja smesti u akumulator
LOAD (100),A	Prenesi sadržaj akumulatora u ćeliju 100

Vidimo da umesto jedne naredbe sada moramo da iskoristimo tri, ali da zato svaku od njih prati samo po jedna adresa. Uz adresu se, međutim, stalno pominje nekakvo slovo A, koje predstavlja ime jednog od registara procesora.

Operativna memorija se obično sastoji od velikog broja (npr. šezdesetak hiljada) ćelija. Osim nje mikroprocesor ima i nekoliko (ili nekoliko desetina) bajtova svoje privatne memorije koji su grupisani u registre: flip-flopovi koji sačinjavaju te registre se nalaze unutar samoga mikroprocesora. Z80 ima četrnaest registara opšte i osam registara specijalne namene kao i nekoliko internih registara koji se koriste za obavljanje nekih specijalnih operacija i koji su programeru nedostupni. Vidljivi registri se obeležavaju slovima A, B, C, D, E, H, L, F, I, SP, PC, IX, IY i R. Registar A nazivamo akumulatorom jer se u njemu akumuliraju rezultati aritmetičkih operacija, dok ćemo imena i namene ostalih registara objasniti kada za to bude došlo vreme. Osmobitni registri A-F se pod određenim uslovima, mogu kombinovati u šesnaestobitne, pa ćemo tako nailaziti na oznaku HL koje označava operaciju nad sadržajima registara H i L posmatranim zajedno.

Neke čudne zestavice

Sledeći termin koji treba da upoznamo je **flag** (ako baš želite da koristite prevode, upotrebite naš „rač. indikator"). Pretpostavljajući da ste razumeli šta je registar, reći ćemo da je flag jednobitni registar. Ovo, bar na prvi pogled, dolazi u kontradikciju sa našom ranijom tvrdnjom da mikroprocesor može da pristupa samo čitavim bajtovima memorije, ali je ta kontradikcija samo prividna: flagovi nisu deo memorije već deo samog procesora i imaju veliku ulogu za njegov normalan rad. Pretpostavimo da treba u ćeliju 100 upisati broj 255 ukoliko se u toj ćeliji ranije nalazila nula ili broj nula ukoliko se u toj ćeliji ranije nalazilo bilo šta drugo. Posmatrajmo program koji bi mogao da obavi tu operaciju

LOAD A (100)	Stari sadržaj ćelije 100 u akumulator
COMPARE A,0	Uporedi sadržaj akumulatora sa nulom
JUMP EQ NULA	Ukoliko je jednak (EQ), idi do mesta u programu koje se zove NULA
LOAD A,0	Broj nula u akumulator
LOAD (100) A	a zatim i u ćeliju 100
HALT	Kraj rada
• NULA	Ovde se dolazi ako je sadržaj ćelije 100 na početku bio 0
LOAD A 255	Broj 255 u akumulator
LOAD (100),A	a zatim u ćeliju 100
HALT	Kraj rada.

Vidimo da je instrukcija COMPARE A 0 bila iskorišćena da se testira sadržaj akumulatora a zatim sma upotrebili naredbu JUMP EQ NULA koja je predstavljala skok samo ako je odgovor na pitanje koje je postavila prethodna naredba bio potvrdan. Kako bi naredba JUMP „znala" kakav je rezultat izvršavanja prethodne naredbe? Jedino tako što je naredba COMPARE A 0 postavila neki flag mikroprocesora u stanje 0 ili 1. Taj flag će biti u stanju jedan ako je odgovor na pitanje da li

je sadržaj akumulatra jednak nuli potvrdan a nule ukoliko je odrečan. Sade se naredba JUMP EQ NULA može pretumačiti rečima idi na deo programa označen sa NULA ukoliko je flag jednak jedinici (ili, kako se to obično kaže ukoliko je flag *setovan)* a produži sa izvršavanjem programa ako nije

Z80 ima više flegova kojima su dodeljena specijalna značenja. O njima ćemo jasno opširno govoriti docnije ze sada je bitno da zamislimo flag kao zastavicu koja se podiže ako je odgovor na neko pitanje potvrdan a spušta ako je odrečan stanje te zastavice docnije možemo da ispitamo i da menjamo tok izvršavanja programa u zavisnosti od njega. Svi flegovi su obično fiktivno povezani u jedan registar koji označavamo slovom F. Smisao ovoga grupisanja je mogućnost čuvanja vrednosti flegova na steku koji predstavlja sledeću oblast našeg interesovanja

Privremena memorija

Obzirom da svaki mikroprocesor ima relativno malo registara često će nam biti potrebno da neke od njih privremeno oslobodimo za obavljanje neke operacije da bismo po njenom završetku, restaurirali njihov ranji sadržaj. U takvoj situaciji bismo mogli da se poslužimo idejom koja je izložena u programu

LOAD (20000) A. Premešta se sadržaj akumulatora u mem. celiju 20000
Deo programa koji koristi akumulator
LOAD A (20000) Obnavljanje ranijeg sadržaja akumulatora.

Nevolja kod ovoga rešenja je što smo morali da pamtimo da je podatak koji smo sačuvali baš u ćeliji čija je adresa 20000 potrošili smo osim toga, osam 6 bajtova za smeštanje LOAD instrukcija. Rešenje istog problema uz upotrebu steka bi glasilo

PUSH A Registar A se čuva na steku
Deo programa koji koristi akumulator
PULL A Obnavlje sadržaj akumulatora.

Sadržaj registra A je, i u ovoj varijanti prepisivan u neku memorijsku ćeliju ali sada mi nismo morali da razmišljamo o njenoj adresi mikroprocesor je opremljen registrom SP *(Stack pointer ili pokazivač steka)* koji sadrži adresu memorijske ćelije koju treba popuniti brojem koji se želje na stek sadržaj ovog registra se automatski povećava ili smanjuje po svakoj operaciji sa stekom tako da primenom PULL instrukcije uvek dobijamo podatak koji smo "zapamtili" primenom prethodne PUSH naredbe. Stek se osim toga intenzivno koristi za realizaciju potprograma i prekida i posvetićemo mu dužnu pažnju u jednom od sledećih poglavlja.

Predstavljanje brojeva

U uvodnom poglavlju smo sa naslovne strane ovih Računara prepisali cenu od 100101100 dinara ali nismo objasnili način na koji smo došli do ovog niza brojeva. da vam je prodavac na kiosku pomenuo cenu od 100101100 dinara našli biste se u velikom čudu! Pre nego što se pozabavimo ovim binarnim brojem moraćemo da posvetimo pažnju običnim, dekadnim brojevima kojima se svakodnevno služimo

Binarni i heksadekadni brojevi

Posmatrajmo, na primer, broj 1986. On je iskazan u pozicionom dekadnom sistemu. Ovo *dekadnom* potiče od toga što je baza sistema broj 10 što znači da oznaka 1986 predstavlja zbir

$$1986 = 6 \cdot 10^0 + 8 \cdot 10^1 + 9 \cdot 10^2 + 1 \cdot 10^3$$
$$= 6 \cdot 1 + 8 \cdot 10 + 9 \cdot 100 + 1 \cdot 1000$$
$$= 6 + 80 + 900 + 1000$$

Što se broja 100101100 tiče on je iskazan u pozicionom binarnom sistemu. Ovo *binarnom* potiče od toga što je baza sistema broj 2 što znači da oznaka 100101100 predstavlja zbir

$$100101100$$
$$= 0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 0 \cdot 2^7 + 1 \cdot 2^8$$
$$= 0 \cdot 1 + 0 \cdot 2 + 1 \cdot 4 + 1 \cdot 8 + 0 \cdot 16 + 1 \cdot 32 + 0 \cdot 64 + 0 \cdot 128 + 1 \cdot 256$$
$$= 0 + 0 + 4 + 8 + 0 + 32 + 0 + 0 + 256$$
$$= 300$$

Vidimo da binarni broj 100101100 i dekadni broj 300 predstavljaju istu stvar izraženu na dva različita načina. Prvi način predstavljanja je pogodniji za računar a drugi za čoveka i postoje

U programu primećujemo jednu interesantnu konstrukciju koja se praksi često koristi LD A H DR L. Ove dva naredbe testiraju vrednost registra HL i u slučaju da je ona jednaka nuli setuju Z flag. Poipunije poređenje šesnaestobitnih registara HL i DE može da se obavi naredbama

```
LD   A H
CP   D
RET  NZ
LD   A L
CP   E
RET
```

Kada izvršite ovaj potprogram Z flag će biti setovan ako je HL=DE. Ukoliko su ova dva registra različita C flag će reći koji je veći

Primer 3: ASCII niz u heksadekadni broj

```
[illegible]        2F   ORG [illegible]
[illegible]        3B   OPT 7
[illegible]        ?B ASCHL
[illegible]        3B   LD HL 0
[illegible]        ?F VRTE
[illegible]        7B   CALL CIFRA   ; SLEDECA CIFRA
[illegible]        ?B   ;            ; ILI KRAJ
[illegible]        9B   ADD HL HL
[illegible]        ?BB  ADD HL HL
[illegible]        11B  ADD HL HL
[illegible]        ?BB  ADD HL HL   ; HL=16*HL
[illegible]        ?3B  ADD A L
[illegible]        14B  LD L A      ;      + CIFRA
[illegible]        15B  JR VRTE
[illegible]        16B CIFRA
[illegible]        ?7B    PREPOZNAVANJE HEX CIFRE
[illegible]        18B  ; NA KOJU POKAZUJE DE
[illegible]        ?1B  LD A (DE)
[illegible]        2BB  CP "0" [illegible]
[illegible]        21B  JP M KRAJ
[illegible]        22B  SUB "0"
[illegible]        23B  CP 10
[illegible]        24B  JP P SLOVO
[illegible]        25F  ; PRONADJENA CIFRA [illegible]
[illegible]        26B  ; DE POKA[illegible]
[illegible]        27B  INC DE
[illegible]        28B  RET
[illegible]        29B SLOVO
[illegible]        3?B  SUB 7
[illegible]        31B  CP 10
[illegible]        32B  JP M KRAJ
[illegible]        33B  CP 1?
[illegible]        34B  JP P KRAJ
[illegible]        35B  ; PRONADJEN[illegible]
[illegible]        36B  ; DE PO[illegible]
[illegible]        37B  INC DE
[illegible]        38B  RET
[illegible]        39B KRAJ
[illegible]        3?B  ; NAIDJEN[illegible]
[illegible]        ?1B  ; NIJE IZ[illegible]
[illegible]        42B    TREBA [illegible]LATI A[illegible]
[illegible]        ?3B    POVRATKA SA STEKA
[illegible]        44B  ; VRATITI SE U PROGRAM
[illegible]        45B    KOJI JE POZVAO ASCHL
[illegible]        46B  POP AF
[illegible]        47B  RET
[illegible]        48B  >
```

Kada poželimo da pišemo mašinske programe koji se neće oslanjati na ulazne i izlazne naredbe bejzika zatrebaće nam mnogo različitih konverzija. Često je na primer potrebno konvertovati niz ASCII znakova koji reprezentuju heksadekadni broj u vrednost pogodnu za računanje. Jedan od jednostavnijih (i ne previše racionalnih) načina da se ovaj problem reši pokazuje i naša slika

Pre pozivanja potprograma ASC—HL treba u registar DE upisati adresu memorijske ćelije koja sadrži prvi znak ASCII niza. Program će konvertovati znakove sva dok ne najde na nešto što nije cifra niti veliko slovo A—F kada će u HL registru biti vraćen odgovarajući broj. Da bi se očuvala jednostavnost nije predviđeno kontrolisanje prekoračenja ukoliko zahtevate od računara da konvertuje niz znakova ABCDEF1 u HL registru će se naći broj &CDEF

Primetimo da glavni program poziva potprogram CIFRA koji kada naide na kraj ASCII niza ne vraća kontrolu glavnom programu naredbom POP AF je zaboravljena adresa povratka pa sledeće RET izaziva povratak u program koji je pozvao ASC—HL. Ovakvo rešenje nije struktuirano i može da izazove zabunu kod nekoga ko čita program ali smo ga iskoristili da ilustrujemo tehniku koju smo u ranijem tekstu pominjali. Ljubiteljima struktuiranog programiranja namenjujemo Primer 4

Dokažite da ste razumeli ovaj program pišući odgovarajuću rutinu koja u HL dovodi vrednost niza ASCII znakova koji su shvaćeni kao ceo dekadni broj. Trebaće vam jedno rutina koja množi HL sa 10 poput sledeće

```
ADD   HL HL
PUSH  HL
ADD   HL HL
ADD   HL HL
POP   DE
ADD   HL DE
```

Primer 4. Broj u ASCII niz

Kao što nam je potrebno da pretvorimo niz znakova u broj tako će nam zatrebati da pretvorimo broj (upisan u HL) u niz ASCII znakova koje ćemo na primer ispisati na ekran. Ovu konverziju možete da povente programu HL—ASC koji je dat na slici. Problem je rešavan sa vrha na dole najpre smo šesnaestobitni broj pretvorili u dve osmobitne i pozvali potprograme koji ih ispisuju. Ti su potprogrami pretvorili osmobitne brojeve u četvorobitne (četiri bita se ponekad zovu *nibl)* i za svaku od ovakvih grupa pozvali potprogram NIBL koji je na neki način inverzan potprogramu CIFRA iz prethodnog primera

Obratimo pažnju na redove 290 i 300 koji glase CALL NIBL RET. Bolji poznavaoci Z80 bi primetili da se u ove dva reda mogu uštedeti dva bajta CALL NIBL stavlja na stek adresu sledeće naredbe (RET) a zatim započinje izvršavanje potprograma. Kada se taj potprogram izvrši njegovo će RET izazvati povratak na adresu koja je upisana na vrh steka tj na naredbu RET koja će ultimo

Sve u svemu, posle CPIR indikator nule će biti setovan ako je podatak pronađen a adresa tog podatka će biti upisana u HL. Ukoliko je BC došao do nule a podatak nije pronađen Z fleg će biti resetovan. Umesto CPIR možemo da koristimo CPI pri čemu smo, kao i ranije sami odgovorni za realizaciju ciklusa korišćenjem PV flega.

Postojanje naredbe CPIR ne ukida potrebu za CPDR (ComPare Decrement&Repeat) i CPD poneko će nam biti neophodno da pretražujemo blok memorije počevši od njegovog kraja. Neka na primer želimo da pronađemo broj &0D u bloku memorije koji se završava na &3000 i pri čemu smo procenili da traženje u memorijskim ćelijama čije su adrese manje od &2000 nema smisla. Napisaćemo

```
        LD      A,&0D
        LD      HL,&3000
        LD      BC,&1000
        CPDR
        JR      Z,found
```

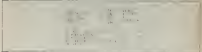
Blokovi registara

Razmena sadržaja osnovnih i alternativnih registara spada strogo uzevši u operacije sa blokovima pa ćemo je još jednom objasniti. Prisećamo se da Z80 osim registarskih parova AF HL DE i BC ima i alternativne registarske parove AF' HL' DE' i BC'. Sa EXX vrlo brzo razmenjujemo HL sa HL' BC sa BC' i DE sa DE' dok se EX AF AF' razmenjujemo sadržaje AF i AF'. Iako ovo izgleda sasvim jednostavno nije lako organizovati program tako da se ovakvim razmenama nešto dobije jeste da ste sa EXX sačuvali tri registarska para ali nemate mogućnosti da dalje koristite njihove vrednosti (posle PUSH AF akumulator će na primer i dalje čuvati stari sadržaj koji je samo prekopiran na stek). U sledećem poglavlju ćemo videti da alternativni registri svoju pravu vrednost dobijaju tek ako je potrebno veoma brzo opsluživanje interapta.

Pomenimo instrukcije EX (SP) HL, EX (SP) IX i EX (SP) IY. Njihovo dejstvo je za nekoga ko je pročitao dosadašnji tekst prilično jasno: broj sa vrha steka se razmenjuje sa brojem koji je upisan u registar HL, IX odnosno IY.

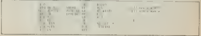
Biblioteka programa

Primer 1: Program koji radi bilo šta



Počećemo od sasvim jednostavnog primera: želimo da računar izvrši sasvim kratak mašinski program koji radi bilo šta — u našem slučaju uvećava vrednost neke memorijske ćelije za 1. Sa slike vidimo da bejzik program sa tastature uzima neki broj a onda ga primenom naredbe POKE upisuje u memorijsku ćeliju čija je adresa &3900. Zatim se primenom naredbe USR poziva mašinski program koji je smešten počevši od adrese &3100 (možete ga jasno smestiti i na mnoga druga mesta) i na kraju ispisuje nova vrednost bajta &3900. Isprobajte ovaj program — pomoći će vam da bolje upoznate vaš asembler i njegove veze sa bejzikom.

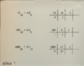
Primer 2. vremenska petlja



Primer 2 predstavlja vremensku petlju — ako vam se ponekad učini da vaš mašinski program radi prebrzo usporite ga ovakvim dodatkom. Trajanje vremenske petlje zavisi od početne vrednosti registra B — ukoliko u njega upišete neki veliki broj (na primer &FF) petlja će potrajati desetak minuta. Upisivanje jedinice u B će sa druge strane izazvati čekanje od svega par sekundi. Ukoliko želite još kraće petlje, postavljajte brojeve manje od &FFFF u HL.

postupci kojima se brojevi iz jednog sistema prevode u drugi. Postupak prevođenja binarnih brojeva u dekadne smo već upoznali ali na pitanje *kako je dobijena brojka 100101100* još nismo odgovorili!

Podelimo najpre broj 300 sa dva. Dobićemo količnik 150 i ostatak 0. Zatim delimo 150 sa 2 i dobijamo 75 i ostatak 0. Sledeće deljenje daje 37 i ostatak 1, zatim 18 i ostatak 1, onda 9 i ostatak 0, 4 i ostatak 1, 2 i ostatak 0, 1 i ostatak 0 i, nazad pri deljenju jedinice sa 2 dobijamo količnik nula i ostatak 1. Prikupimo sve ostatke i napišimo ih obrnutim redom počevši od poslednjeg. Dobijamo broj 100101100, tj. predstavu broja 300 u binarnom sistemu. Radi vežbe možete da pogledate sliku 1 na kojoj je tabelarno prikazana ova i nekoliko drugih konverzija dekadnih brojeva u druge brojne sisteme.



slika 1

Dekadni	Binarni	Heksadekadni
0	0000	&0
1	0001	&1
2	0010	&2
3	0011	&3
4	0100	&4
5	0101	&5
6	0110	&6
7	0111	&7
8	1000	&8
9	1001	&9
10	1010	&A
11	1011	&B
12	1100	&C
13	1101	&D
14	1110	&E
15	1111	&F

slika 2

U poslednjem primeru sa slike 1 smo konvertovali broj 1000 u tzv. heksadekadni sistem tj. brojni sistem čija je osnova 16. Dobili smo ostatke 3, 14 i 8, što znači da bi rezultujući broj trebalo napisati kao 3148. Ovo naravno ne bi valjalo jer broj 3148 može da se sastavi i od ostataka 3, 1, 4 i 8. Zato bismo ovaj broj morali da napišemo kao *(3)(14)(8)* ili, ako želimo da uštedimo malo pisanja, kao *3E8*. Odakle se pojavilo slovo E? Za pisanje binarnih brojeva su dovoljne cifre 0 i 1 što znači da bi se neki vanzemaljac koji potiče iz kompjuterskog sveta u kome se operiše samo sa binarnima veoma začudio kada bi video naše cifre 2, 3, ... 8, 9. Za heksadekadne brojeve treba imati 16 cifara, tj. šest više nego za dekadne. Mogli smo naravno da smislimo neke sasvim nove znakove koji bi predstavljali ova cifre ali bi takav postupak zahtevao predizajniranje tastature i generatore karaktera naših računara. Daleko je jednostavnije zameniti cifre 10, 11, 12, 13, 14, i 15 slovima A, B, C, D i E prema tabeli sa slike 2.

Na slici 2 primećujemo i jednu interesantnu osobinu heksadekadnih brojeva. Posmatrajmo na primer broj 1000 napisan binarno i heksadekadno

1000 = 0011 1110 1000
= 3 E 8

Vidimo da dok za prevođenje dekadnih u binarne brojeve treba primenjivati relativno komplikovan algoritam konverzija binarnih u heksadekadne brojeve i obratno predstavlja jednostavan postupak koji se brzo nauči napamet. To je upravo i razlog za uvođenje heksadekadnih brojeva — oni su izvanredna zamena za binarne (koji su kao što rekosmo jedini prihvatljivi za računar) a nisu predugački da bi ih imalo smisla pisati i pamtiti (mnogo je lakše zapamtiti da se bejzik program kod galaksije smešta u memoriju počevši od adrese 203A nego pamtiti broj 0011100000111010!)

Ostalo je da razrešimo još jednu dilemu: kako ćemo prepoznati u kom je sistemu predstavljen neki broj. Binarni brojevi su obično toliko dugački da njihovo prepoznavanje ne predstavlja problem, nevolja je što naiđavši na broj 3000 ne znamo da li se radi o dekadnoj ili o heksadekadnoj konstanti. Matematičari su uobičajili da se iza svakog broja piše oznaka brojnog sistema kao indeks npr. 3000_{10} ili 3000_{16}. Ovakav način pisanja je za kompjuterske primene sasvim nepogodan jer računari ispisuju slova u precizno formatiranim redovima. Zilog je zato uveo konvenciju da se iza broja ne piše nikakvo slovo ako se radi o dekadnoj konstanti, da se piše slovo B ako se radi o binarnom broju a slovo H ako se radi o heksadekadnom. Tako bismo mogli da napišemo 3E8H, 1000 ili 1111101000B. Iako je ovakav način označavanja pogodan za programiranje u mašinskom jeziku autori bejzik interpretatora se trude da ga izbegnu jer može da dovede do zabune: da li na primer protumačiti naredbu

IF A=2000H=0 kao
IF A=2000 THEN H=0 ili kao sintaksnu grešku
IF A=2000 H THEN=0?

Kako se kod mnogih kućnih računara paralelno koristi bejzik i mašinski jezik, neprijatno je zahtevati od korisnika da u bejziku koristi jedne a u mašinskom (zapravo asemblerskom) razlikama između ova dva pojma govorimo u poglavlju 4) druge oznake. prihvaćeno je da se ispred dekadnih brojeva ne stavlja nikakav prefiks, da se ispred heksadekadnih stavlja oznaka & a ispred

binarnih %. Na taj način bismo cenu ovih „Računara" mogli da napišemo kao 300, &120 ili %100101100 dinara. U „spektrumovim" asemblerima kao što je *Devpac* je uobičajeno da heksadekadnim brojevima prethodi oznaka # (number sign ili u domaćem hakerskom žargonu „taraba"). Obzirom da ova oznaka u mnemonici 6502 i nekih drugih mikroprocesora označava neposredno adresiranje, za nas je bilo apsolutno nemoguće da prihvatimo ovakvu konvenciju bez obzira na problem koji one može da izazove kod dela čitalaca.

Operacije sa binarnim brojevima

U osnovnoj školi se uči sabiranje i oduzimanje dekadnih brojeva pa je neophodno da se u ovoj osnovnoj školi mašinskog programiranja posvetimo operacijama sa binarnim veličinama. U osnovi, ako znate da sabirate dekadne brojeve, znate da sabirate i binarne, heksadekadne ili brojeve u bilo kom drugom sistemu. Činjenica da nešto znate, naravno, još ne znači da to radite tačno i brzo: ukoliko želite da steknete malo rutine (koja, inače, nije previše nužna za pisanje mašinskih programa) moraćete da posvetite određeno vreme uvežbavanju.

Izaberite neka dva relativno mala dekadna broja i saberite ih. Ako su to brojevi 7 i 11, dobićete zbir 18. Sada pretvorite 7 u binarni broj 111 i 11 u binarni broj 1011. Potpišite ova dva binarna broja i pokušajte da ih saberete.

Jedan i jedan su 2, ali se to dva piše kad 10 što znači da nulu pišete a jedan pamtite. Taj jedan što ste pamtili plus i iz broja 111 plus 1 iz 1011 daju 3 odnosno 11: pišete 1 i prenosite 1. Taj jedan plus 1 iz 111 plus 0 iz 1011 daju 2 tj. 10: pišete nulu i pamtite 1. Saberete, najzad, to jedan sa jedan iz 1011 i dobijate 2 tj. 10 koje napišete: rezultat je 10010. Pretvorite ovaj rezultat u dekadni broj

$$10010=0\cdot2^0+1\cdot2^1+0\cdot2^2+0\cdot2^3+1\cdot2^4=18$$

Rezultat je 18 što je, kao što ste verovatno i očekivali, isto što i 7+11. Na vama je da isprobate ono što ste upravo naučili na drugim primerima što je, naravno, ujedno i lepa škola pretvaranja binarnih brojeva u dekadne i obrnute operacije.

Oduzimanje biste mogli da obavljate na sličan način: umesto prenosa koristili biste „pozajmicu". Želimo međutim da ukažemo na jedan drugi način oduzimanja koji je daleko bliži mikroprocesorima sa kojima radimo. Poznato nam je da je 12−7 ISTO ŠTO i 12+(−7) što znači da se oduzimanje svodi na sabiranje sa negativnim brojem. Došlo je, dakle, vreme da upoznamo negativne binarne brojeve.

Broj −111 je, naravno, isto što i −7 ali je −111 broj vrlo nepogodan za pamćenje u memoriji računara: osim jedinice i nule u njemu se pojavljuje i neka crtica koja predstavlja negativan predznak. Ako pretpostavimo da se svaki broj upisuje u jedan bajt memorije, mogli bismo da se dogovoramo da prvi (ili poslednji ako, kad što je uobičajeno, bitove brojite adesnaj) bit tog bajta u sebi kodira znak tako da će se broj 7 pamtiti kao 00000111 a broj −7 kao 10000111. Ovakvo predstavljanje brojeva se međutim pokazalo nepogodnim za mnoge primene, pa je izmišljeno nešto bolje — pamćenje komplementa broja. *Prvim komplementom* broja 7 (00000111) nazivamo binarni broj koji dobijamo kada umesto svake nule u tom broju stavimo jedinicu a umesto svake jedinice nulu: 00000111 tako postaje 11111000. Dodajući jedan na ovaj broj dobijamo 11111001 što je *drugi* ili *potpuni komplement* broja 7 i, ujedno, zgodan način da se predstavi broj −7.

Pokušajmo sada da izračunamo koliko je 12−7=12+(−7). Broj 12 binarno predstavljamo kao 00001100, a broj −7 kao 11111001. Sabiranjem ova dva broja dobijamo 100000101. Vidimo da rezultat ima 9 bita, kako u memorijsku ćeliju može da se smesti najviše 8, deveti bit će biti odbačen tako da će rezultat biti 00000101 ili, dekadno posmatrano, 5 što je, kad nakom čarolijom isto što i 12−7. Iz ovog primera i svih primera koje ćete sami zamisliti da se zaključiti da je predstavljanja negativnih brojeva pomoću potpunog komplementa celishodno rešenje koje svodi oduzimanja na sabiranja sa negativnim brojem.

U prethodnom primeru smo, međutim, mirna duša odbacili poslednji (rekli smo da se bitovi broje sa desna strane) bit rezultata i dobili korektno rešenje: može se pretpostaviti da bi kod nekog drugog primera takvo nonšalantno ponašanje dovelo do greške. Da stvar bude još gora, greška je moguća čak i kada nema nikakvog prekoračenja. Da smo, na primer, pokušali da saberemo 121 i 93, tj. 01111001 i 01011101, dobili bismo

```
  01111001
+ 01011101
----------
  11010110
```

Obzirom da je poslednji bit rezultata jedan, radi se o negativnom broju koji, pretvoren u dekadni, daje 11010110=−00101010=−42. Jasno je da −42 nikako ne može da bude rezultat sabiranja brojeva 121 i 93. Da nam se u budućnosti ne bi događale ovakve greške, naučićemo tri pravila:

1. U memorijsku ćeliju mogu da se smeštaju označeni ili neoznačeni brojevi. U jedan bajt (osam bita) mogu da se smeste označeni brojevi između −128 i +127 *ili* neoznačeni brojevi između 0 i 255.

ABRAKADABRA upotrebićemo čarobnu reč LDIR (LoaD, Increment&Repeat) i — ništa više! Nema labela, nema ciklusa, nema IF-a, nema potrebe za flegovima.

Kako radi instrukcija LDIR, Mogli biste da je zamenite sa

```
ciki    LD     (DE),(HL)    kada bi ovako nešto postojalo
        INC    HL
        INC    DE
        DEC    BC
        CP     BC,0         kada bi ovakvo nešto postojalo
        JR     NZ,cikl
```

Već vas čujem kako kažete: baš je lepo iskoristiti neku instrukciju a onda u komentaru napisati „kada bi ovakva instrukcija postojala". Pokušajmo, vežbe radi, da napišemo isti program uz korišćenje isključivo postojećih instrukcija (zamena LDIR neće biti baš ekvivalentna jer ova instrukcija utiče na PV, N i H flegove, ali koga za njih briga?)

```
        PUSH   AF           ; čuva A i flegove
cikl    LD     A,(HL)       ; sledi zamena za LD (DE),(HL)
        LD     (DE),A
        INC    HL           ; modifikacija brojača
        INC    DE
        DEC    BC
        LD     A,B          ; simulacija CP BC,0. Zašto?
        OR     C
        JR     NZ,cikl
        POP    AF
```

Upoznajmo sada instrukciju LDI koja je, kao što možete da pretpostavite, ekvivalentna sa LDIR ali ne izaziva ponavljanje operacije: najpre se izvrši nešto poput LD (DE),(HL), povećaju se HL i DE za po jedan a BC se umanji za 1. Tada se BC uporedi sa nulom pa se rezultat smesti u Parity/Overflow fleg (???) i to tako da je fleg resetovan ukoliko je BC postalo nula a setovan u svim ostalim slučajevima. Obzirom da se LDI retko koristi, nećemo joj posvećivati više pažnje.

Dejstvo instrukcije LDDR ćemo upoznati posmatrajući njenu simulaciju uz obavezno korišćenje nepostojećih instrukcija (tako je zabavnije, čini vam se da LDDR za nešto i služi!)

```
cikl    LD     (DE),(HL)
        DEC    HL
        DEC    DE
        DEC    BC
        CP     BC,0
        JR     NZ,cikl
```

Jedina razlika između LDDR i LDIR je što se primenom prve blok kopira od kraja ka početku a kod druge od početka ka kraju. Da bismo, dakle, rešili problem sa početka ovoga poglavlja koristeći LDDR, morali bismo da napišemo

```
        LD     HL,&30FF
        LD     DE,&32FF
        LD     BC,&100
        LDDR
```

Uz LDDR ćemo, jasno, pomenuti LDD: instrukciju koje bi trebala da vam bude jasna ako ste shvatili razliku između LDI i LDIR.

Zbog čega su konstruktori Z80 obezbedili LDD i LDDR kada su već postojale LDI i LDIR? Najsigurniji ćete odgovor dobiti ako ih pitate, ukoliko vam se na putuju do Amerike, poverujte u naše nagađanje: ove instrukcije verovatno ne zauzimaju mnogo prostora u mikrokodu a znate procesori se reklamiraju i navođenjem broja instrukcija koje prepoznaju!

Pretraživanje blokova

LDI, LDIR, LDD i LDDR su instrukcije za pomeranje blokova memorije pa nam je ostalo da upoznamo instrukcije koje omogućavaju njeno pretraživanje. Cilj pretraživanja je jasno: pronaći neki broj u jednoj od sukcesivnih memorijskih lokacija. Taj broj se smešta u akumulator, dok se u HL upisuje adresa početka bloka memorije koji se pretražuje. U BC treba upisati maksimalni broj bajtova koji se pretražuju: ako ste sasvim (ali baš sasvim) sigurni da će podatak pre ili kasnije biti pronađen, možete da napišete LD BC,&FFFF ili, uz malo smisla za humor, LD BC,0. Iza svega toga pišete instrukciju CPIR (ComPare, Increment&Repeat) koja bi mogla da se napiše i kad

```
cikl    CP     A,(HL)
        JR     Z,izlaz
        INC    HL
        DEC    BC
        CP     BC,0
        JR     NZ,cikl
        RES    Z            ; zar ne bi bilo lepo imati instrukciju
                            ; koja resetuje indikator nula
```

adresu instrukcije na koju će se preći po nailasku na RET. U principu bismo mogli da napišemo LD HL,&3000· PUSH HL, a zatim RET, mikroprocesor bi se, vraćao na instrukciju koja je upisana u memorijsku ćeliju &3000 bez obzira na činjenicu da prethodno nije izvršeno nikakvo CALL, tako smo, pravde to neme mnogo smisla, simulirali JP &3000!

Daleko je problematičnija situacija u kojoj smo u potprogramu koji je pozvan sa CALL nehotice upotrebili neko PUSH koje nije imelo svoje POP. Naišavši na RET mikroprocesor slepo uzima dva bajta sa steka i smešta ih u PC ne znajući da se ne radi ni o kakvoj povratnoj adresi već o podatku koji ima neki drugi smisao. Izvršavanje programa će se u tom slučaju nastaviti od nekog nepredviđenog mesta što može da ima katastrofalne posledice po program u memoriji! Sve u svemu, **ako se planira povratak iz potprograma**, upotreba steka mora da bude „čista": koliko podataka stavimo na stek toliko moramo i da uzmemo i tako u svakoj grani potprograma!

Zašto smo rekli „ako se planira povratak"? Kada u bejziku izvršimo GOSUB **moramo** da izvršimo i RETURN, inače se (nama nevidljiv) stek za pozive potprograma polako puni pa će jednom biti prijavljena greška tipa 'Too many GOSUBs'. Ne bi imalo nikakvog smisla učiti mašinski jezik ako bi i on imao slično ograničenje: ako u nekoj grani potprograma konstatujemo da se nema smisla vraćati u glavni program, jednostavno POP AF će skloniti sa steka svaki trag da je neki potprogram uopšte pozivan. Novi sadržaj akumulatora i flag registra u tom slučaju treba ignorisati.

Često je potrebno da se potprogramu prenesu neka vrednosti koje će obrađivati i da se u glavni program vrate rezultati te obrade. Argumente potprograma je najjednostavnije upisati u neke od registara Z80 ih i onako ima dosta. Ukoliko je potrebno više argumenata, možete ih upisati u neke memorijske ćelije sa tačnim adresama odakle će ih potprogram „pokupiti" u tom slučaju potprogram **ne sme** da poziva samog sebe, ali su vam ovakve tzv. rekurzije za sada sasvim nepotrebne. Moguće je najzad smestiti sve argumente na stek, a onda izvršiti CALL. Potprogram tada najpre mora da prenese adresu povratka u neki od registara (npr. sa POP IX), zatim da obradi argumente postepeno ih skidajući sa steka, da smesti rezultate na stek i da konačno izvrši JP (IX). Sve u svemu, prilično komplikovano.

Specijalni slučaj instrukcije CALL je **RST (ReSTart)**: dok CALL zajedno sa adresom potprograma zauzima uobičajena tri bajta, RST zajedno sa svim potrebnim podacima zauzima samo jedan. Kako je to moguće? Sa RST se može pozvati samo jedan od osam potprograma koji se nalaze na fiksnim memorijskim adresama &0, &8, &10, &18, &20, &28, &30 i &38. Sve se ove adrese nalaze u ROM-u i predstavljaju početka nekih vrlo važnih i često korišćenih potprograma. Umesto RST &10 možete, naravno, uvek da koristite i CALL &10 ukoliko vam utrošak dva bajta nije važan, instrukciju RST možete i da zaboravite.

U ROM-u vašeg računara se nalazi mnogo korisnih potprograma koji vas oslobađaju briga o pisanju po ekranu, skaniranju tastature, generisanju zvuka i slično: zar nije lakše napisati CALL KEY-SCAN i u akumulatoru dobiti kod pritisnutog tastera nego ispitivati dirke jednu po jednu? Da biste, međutim, napisali CALL KEY-SCAN morate da znate gde se tačno u ROM-u nalazi rutina koja skanira tastaturu (ako je ona smeštena počevši od adrese &028F kao kod „spektruma", na početku vašeg programa ćete napisati KEY-SCAN EQU &028F), kako treba pripremiti njen poziv i gde su smešteni rezultati. Takve podatke možete da nađete u knjigama tipa „The Complete Spectrum ROM Disassembly". Nabavite ako je to ikako moguće knjigu u kojoj je opisan ROM vašeg računara i videćete da će one postati najkorisnije reference u praktičnom radu!

Operacije sa blokovima

Koristeći do sada naučene instrukcije ne bi vam bilo teško da, na primer, napišete program koji će prepisati segment memorije od &3000 do &30FF (zaključno) na memorijske adrese &3200—&32FF.

Program koji smo upravo napisali korektno rešava zadati problem i, kada ga poredimo sa odgovarajućom bajzik rutinom, predstavlja pravog Brzog Gonzalesa. Konstruktori Z80 su nam međutim omogućili da napišemo sasvim sličan program koji će se sastojati od samo četiri instrukcije, biti dosta brži i daleko jasniji na prvi pogled. Uvođenje novih (i u suštini nepotrebnih) instrukcija je jasno mač sa dve oštrice jer je početniku često teško da odvoji bitno od nebitnog. Zato bismo vam savetovali da pri prvom čitanju ovog umetka jednostavno preskočite ostatak ovoga poglavlja zapamtivši jedino da postoje moćne instrukcije za pomeranje i pretraživanje blokova memorije, kada vaše poznavanje mašinskog jezika kroz praktičan rad bude uznapredovalo neće biti kasno da naučite više o ovim „stvarčicama".

Pomeranje blokova

Instrukcije za pomeranje memorijskih blokova koriste **sve** registre opšte namene osim paradoksalno akumulatora. Registar HL **uvek** sadrži adresu početka bloka, registar DE njegovo odredište (destination), dok u BC treba upisati broj bajtova koji se prenose. U primeru koji smo deli na početku ovoga poglavlja podeci koji se premeštaju počinju od memorijske ćelije &3000 što znači da će naš mali program početi sa LD HL,&3000. Podeci se premeštaju u memorijski blok koji počinje od &3200: LD DE,&3200. Potrebno je premestiti &100 bajta dakle LD BC,&100. Umesto

2. Pri sabiranju neoznačenih brojeva može da se pojavi prenos iz najvišeg razreda koji nazivamo *carry*. Pojava ovoga prenosa pri računanju implicira postojanje greške tj. pokušaj da se saberu dva broja čiji je zbir veći od 255.

3. Pri sabiranju označenih brojeva prenos treba ignorisati. Moguće je međutim da se pojavi prekoračenje tj. greška u računu koji nazivamo *overflow*. Mikroprocesori Z80 i 6502 omogućavaju programeru da proveri postojanje greške (koja čini rezultat u akumulatoru besmislenim) i preduzme neku akciju ukoliko je ona nastupila.

Binarno množenje se realizuje uz pomoć jedne nove operacije koju nazivamo *šiftovanje* i običnog sabiranja. Kako ćemo se na problem šiftovanje vratiti već u sledećem poglavlju množenju sada nećemo posvećivati posebnu pažnju. Što se deljenja tiče, možete ga obavljati slično deljenju običnih brojeva.

Logičke operacije

Programeri koji svoje remek-dela isključivo na višim programskim jezicima možda smatraju da su sabiranje, oduzimanje, množenje, deljenje i eventualno stepenovanje jedine operacije koje računar može da obavi. Ukoliko se međutim upoznate sa nekim ko u životu piše jedino mašinske programe, verovatno ćete saznati da računari umeju da sabiraju (eventualno i da oduzimaju), da se sa množenjem slabo snalaze, da im deljenje predstavlje veliki problem, ali da su im operacije čudnih imena *konjunkcija*, *disjunkcija*, *negacija* i *šiftovanje* najjača strana.

Logičko "I"

Konjunkcija je latinski naziv za logičku I funkciju koja je definisana sledećom tablicom

[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]

[illegible]	[illegible]	[illegible]
0	0	0
0	1	0
0	0	1
1	1	1

slika 3

Pogledajmo smisao ova tablice na jednom primeru: neka je u memorijskoj ćeliji čiji je redni broj 100 upisana konstanta 120 (ili %1111000) a u ćeliji 121 konstanta 72 (%1001000). Naš računar izvršava program poput sledećeg:

```
LOAD A (100)      U akumulator se dovodi broj %1111000
AND A (101)       Sadržaj akumulatora AND sadržaj ćelije 101
                  rezultat se smešta u akumulator
LOAD (102) A      Vrednost akumulatora se premešta u ćeliju 102
HALD              Kraj rada
```

Pitanje je naravno koji broj dolazi u memorijsku ćeliju 102. Potpišimo brojeve %1111000 i %100100 jedan ispod drugog i primenimo na njih tabelu sa slike 3 gledajući nezavisno svaki njihov bit. Dobijamo

```
    1111000
AND 0100100
-----------
    0100000=32
```

Primećujete li vrlo bitnu razliku između sabiranja ili bilo koje aritmetičke operacije koju ste do sada poznavali i operacije I? Kada sabirate višecifreni broj, to radite zdesna nalevo vodeći računa o eventualnom postojanju prenosa u viši razred. Kod logičkog I gledate svaki par potpisanih bitova bez obzira na poziciju u broju, pa biste operaciju mogli da obavite i sleva nadesno pa čak i ukoliko vam je takvo nešto potrebno, počevši iz sredine broja.

Uz prethodni primer i još nekoliko koje sami izmislite (pokažite, na primer, da je 12 AND 28=12 ili da je &7F+&80=%0) razumećete pravila uz pomoć kojih se obavlja logička operacija I, ali nećete shvatiti njenu svrhu. Pretpostavimo, na primer, da je u memorijskoj ćeliji čija je adresa 100 upisan neki broj i da nam je iz nekog razloga potrebno da sedmi (poslednji) bit tog broja postavimo na nulu, a da ostale bitove ne promenimo. Već nam je poznato da mikroprocesor može da pročita i u memoriju upiše samo čitav bajt što znači da bi bez primene logičkih operacija manipulacija pojedinim bitovima bila praktično nemoguća. Uz poznavanje funkcije I napisaćemo jednostavan program:

```
LOAD A (100)        Dovođenje sadržaja ćelije 100 u akumulator
AND A %1111111      Resetovanje sedmog bita
LOAD (100) A        Smeštanje rezultata u memoriju
```

Pre nego što vidimo kako radi ovaj program primetimo razliku između njega i prethodnog primera u kome je druga linija glasila AND A (101). Namerno smo istekli zagrade koje se ne pojavljuju u primeru koga upravo posmatramo. Dok AND a (101) znači *primeni logičku operaciju AND (I) na sadržaj akumulatora i* **sadržaj memorijske ćelije** *čiji je broj 100* dotle AND A %01111111 znači *primeni logičku operaciju AND (I) na sadržaj akumulatora i* **broj %01111111**. Zagrade dakle označavaju da je u okviru naredbe navedena adresa memorijska ćelija u kojoj se nalazi traženi podatak dok njihovo izostavljanje označava da je u okviru naredbe naveden baš taj podatak.

Pretpostavimo da se u ćeliji 100 nalazio broj &9A, odnosno %10011010. On je najpre doveden u akumulator a zatim je na njega i broj %01111111 primenjana operacija I. Da vidimo šta se dobija:

```
    10011010
AND 01111111
------------
    00011010
```

Vidimo da se rezultat razlikuje od početnog podatka &9A samo u poslednjem bitu koji je posle primene AND operacije postao nula. Ovakav rezultat se mogao i očekivati: broj %01111111 se sastoji od samih jedinica od kojih svaka kada je logički pomnožimo (operacija I se zove i logičko množenje) sa bilo čim drugim daje kao rezultat to nešto drugo. Jedino vodeća nula kada je logički pomnožimo sa bilo čim daje kao rezultat opet logičku nulu!

Osim postavljanja bitova na nulu operacija AND ima još jednu značajnu ulogu koju nazivamo maskiranjem. Pretpostavimo da smo memorijsku ćeliju 100 namenili za smeštanja nekih podataka koji imaju samo po dva bita. To znači da desni („nulti") bit te ćelije kao i bit do njega („prvi bit") predstavljaju jedan podatak, sledeća dva bita drugi i tako dalje:

DDCCBBAA

Da bi ovakva koncepcija uopšte imala smisla, moramo da omogućimo čitanja i promenu bilo kog od podataka AA, BB, CC i DD. Što se postavljanja nekih bitova na nulu tiče nema problema. Kako međutim da pročitamo podatak BB?

```
LOAD A,(100)      Sadržaj ćelije 100 u akumulator
AND A %00001100   Maskiraju se bitovi 2 i 3
```

Ako se pre izvršenja ove dve naredbe u ćeliji 100 nalazio broj %01011001 posle njihove primene ćemo u akumulatoru dobiti

```
    01011001
AND 00001100
------------
    00001000
```

Primenom takozvanog „maskiranja" smo dakle uspeli da izdvojimo tražena dva bita iz jednobajtnog broja i dovedemo ih u akumulator. Oni se na žalost ne nalaze baš na pravom mestu (odgovaralo bi nam da se u akumulatoru nalazi broj 0000010) ali ih temo možemo dovesti primenom šiftovanja koje ćemo za trenutak upoznati. Imamo međutim još jedan problem: za sada nemamo načina da postavimo neki bit (ili neke bitove) u stanje logičke jedinice. Za to će nam poslužiti disjunkcija tj. logičke operacija ILI koju često nazivamo *logičko sabiranje* i obeležavamo sa OR.

Logičko "ILI"

Na slici 4 je date tablica istinitosti disjunkcija koja je, kao što vidimo, sušta suprotnost konjunkciji koju smo već upoznali. Pogledajmo kako operacija ILI deluje na primeru:

```
   1111000
OR 0100100
-----------
   1111100=124
```

Par strana unazad smo uzgred budi rečeno videli da se primenom logičkog I na ista dva broja dobija 32. Operacija I daje logičku jedinicu u rezultatu samo na mestima gde se **u oba** operanda javlja logička jedinica dok ILI daje jedinicu svuda gde **se u bilo kom** operandu nalazi jedinica (pročitajte prethodnu rečenicu još par puta). Kao takva operacija ILI je vrlo zgodna za postavljanja bitova u stanje 1.

Govoreći o operaciji I napisali smo *mali* program koji postavlja poslednji bit memorijske ćelija 100 na nulu, pa ne bi bilo loše da ga dopunimo programom koji isti bit postavlja na jedinicu:

```
LOAD A,(100)      Dovođenja sadržaja ćelije 100 u akumulator
OR   A %10000000  Setovanje sedmog bita
LOAD (100),A      Smeštanje rezultata u memoriju
```

smeštajući nulu (sadržaj akumulatora) u memorijsku ćeliju koju pokazuje HL i umanjujući sadržaj ovog registra za 1 kako bi pokazivao sledeću ćeliju koju treba brisati. Sledi naredba DJNZ PETLJA koja pri svakom izvršavanju umanjuje sadržaj B za jedan a onda izaziva skok na labelu PETLJA **samo ako je** sadržaj B različit od nula. Kada čitav segment memorija bude obrisan B će doći do nula i DJNZ će biti preskočeno mikroprocesor nailazi na RET i vraće kontrolu bejziku. Kako je instrukcija DJNZ predviđena za realizovanja ciklusa one uvek izaziva skok unazad što znači da ciklusi mogu da budu dugi do 255 bajta dok bi korišćenjem DEC B JR NZ CIKLUS morali da budu duplo kraći.

Potprogrami i stek

Došlo je vrema da se podrobnije upoznemo sa stekom, koji smo već više puta pominjali. Stak je rekli smo područja memorije koje se koristi za privremeno odlaganje podataka koji će u bliskoj budućnosti biti potrebni. Na stek primanom instrukcija PUSH možemo da smestimo bilo koji od **registarskih parova** AF, DE, HL i BC kao i specijalna registre IX i IY. Pogledajmo najpre kako radi PUSH HL:

```
DEC  SP
LD   (SP),H
DEC  SP
LD   (SP),L
```

Ovaj program ne bi radio kada biste ga otkucali jer Z80 ne poznaje instrukciju LD (SP),reg. I pored toga on rečito pokazuje sve što se dešava: šesnaestobitni registar biva upisan u memorijske lokacija (SP-1) i (SP-2) dok sam SP biva umanjen za dva tako da pokazuje na poslednji bajt na staku. Kada nam nekoliko instrukcija docnije, buda potrebno da vratimo podatak sa steka u HL izvršićemo POP HL odnosno

```
LD   L,(SP)
INC  SP
LD   H,(SP)
INC  SP
```

Šta se dešava ako napišemo PUSH HL, PUSH DE? Na stek najpre biva smešten registarski par HL a onda DE što znači da za vraćanja podataka u registre treba koristiti POP DE, POP HL **a ne** POP HL, POP DE. Zbog toga se stak naziva i LIFO strukturom (LIFO=last in first out): poslednji podatak koji stavimo na stek će biti prvi koji ćemo primanom POP instrukcija pročitati.

Jasno je da izvršavajući POP DE mikroprocesor ne može da zna da li su dva bajta koja pronađe na steku nakada bila u registru DE: oni su se tamo mogli naći i posle PUSH HL. To znači da kombinacijom PUSH HL, POP DE možemo da koristimo umesto LD DE, HL instrukcija koju Z80 ne podržava. Nešto je međutim brže napisati LD D,H; LD E,L sa istim efektom.

Kada u bejziku napišete GOSUB 1000 računar će početi da izvršava potprogram koji počinje od linije 1000. Kada u daljem radu naiđe na RETURN vratiće se izvršavanju segmenta programa koji se nalazi iza GOSUB 1000. Da bi to uradio on na neki način mora da „zna" da je do linije 1000 stigao uz jedan poziv potprograma (inače će RETURN izazvati grešku „RETURN without GOSUB") i da „zapamti" gde se nalazila naredbe koja je pozvala potprogram. Ovi podaci se zaista i pamte ali su za prosečnog programera nebitne njihove lokacija.

Z80 je opremljen instrukcijom CALL koja je potpuni ekvivalent GOSUB: razlika je jedino u tome što se iza GOSUB piše broj programske linija a iza CALL memorijska adresa ćelija u koju je upisana prva instrukcija potprograma (uz korišćenja asemblera će se, jasno, iza CALL naći ime labela). Za povratak u glavni program koristi se RET, potpuni ekvivalent RETURN.

Da bi stvar bila još lepša konstruktori Z80 su obezbedili uslovni CALL: CALL Z PROBA će izvršiti potprogram označen labelom PROBA samo ako je Z(ero) flag setovan dok će se ako nije izvršiti sledeće instrukcija u programu. Umesto Z možete da koristite NZ, C, NC, PO, PE, P ili M baš kao i kod uslovne JP instrukcije. Za povratak iz potprograma možete da koristite RET ali i RET C, RET NC, RET Z itd. ukoliko je uslov ispunjen kontrola se vraće glavnom programu a ako nije nastavija se sa izvršavanjem naredbe iza RET (uslov).

Da biste znali sve o potprogramima reći ćemo da nema prepreke da u toku izvršavanja nekog potprograma pozovete novi potprogram a iz ovoga sledeći i tako do prilične dubina.

Kada bismo ovima zaključili poglavlja o potprogramima znali biste koliko i bejzik programer koji mirno koristi GOSUB ... RETURN ne razmišljajući o tome kako ove strukture funkcionišu. Mašinski programer **mora da zna više**, pa ćemo pokušati da prikažemo delovanje instrukcija CALL i RET.

Kada naiđe na RET Z80 mora da zna gde da se vrati (ne bi bilo loše kada bi znao i da li je potprogram stvarno pozvan ili se do RET-a stiglo greškom ali to nije moguće — mikroprocesor kao što je Z80 i onako ne bi mogao da prijavi (RETURN without GOSUB). Pošto se povratak vrši na instrukciju koja je upisana u neku memorijsku ćeliju pri nailasku na RET su potrebna samo dva bajta adresa te ćelija. Gda bi ta dva bajta mogla da budu smeštena? Pa jedini ozbiljan kandidat može da bude stek.

CALL nnnn je otprilike ekvivalentno sa PUSH PC, JP nnnn a RET sa POP PC (jasno je da instrukcija PUSH PC i POP PC pod tim imenima ne postoje). Pri svakom pozivu potprograma na stek dakle bivaju smeštena dva bajta koja posmatrana kao šesnaestovni broj predstavljaju

Slično tome C predstavlja situaciju u kojoj je indikator prenosa setovan a NC situaciju u kojoj je restovan P označava pozitivan a M negativan znak broja (N fleg resetovan odnosno setovan) dok PE i PO označava setovan odnosno resetovan indikator parnosti/prekoračenja Za početak će nam kao što vidimo na slici 13 biti dovoljni Z i N flegovi program koji dajemo naime u memorijsku lokaciju REZ upisuje jedinicu ako je broj u ćeliji POD pozitivan nulu ako je jednak nuli a 255 ukoliko se radi o negativnom broju (niste zaboravili da mikroprocesor kada se radi o jednobajtnim rečima, smatra brojeve 0—127 pozitivnim a 128—255 negativnim uz korišćenje potpunog komplementa?)

```
3000                 ORG   &3000
3100          POD    EQU   &3100
3101          REZ    EQU   &3101
3000 3A0031          LD    A,(POD)
3003 FE00            CP    0
3005 2004            JR    NZ,NIJE_NULA
3007 320131          LD    (REZ),A
300A C9              RET
300B          NIJE_NULA
300B F2143D          JP    P,POZITIVAN
300E 3EFF            LD    A,&FF
3010 320131          LD    (REZ),A
3013 C9              RET
3014          POZITIVAN
3014 3E01            LD    A,1
3016 320131          LD    (REZ),A
3019 C9              RET
```

slika 13

Relativni skokovi

Govoreći o raznim vrstama adresiranja naučili smo da Z80 podržava relativno adresiranje samo kod nekih instrukcija uslovnog i bezuslovnog skoka. Šta beše relativno adresiranje? Ponovo ćemo ga upoznati posmatrajući instrukciju JR PC+&20 (Jump Relative &20 bytes) Pretpostavimo da se ova instrukcija nalazila u memorijskim ćelijama &1000 i &1001 (videćemo da instrukcije relativnog skoka za razliku od instrukcija apsolutnog zauzimaju samo dva bajta memorije) Mikroprocesor je najpre iz memorije uzeo kod instrukcije JR (&18) i prepoznao ga a zatim pročitao adresu &20 U tom trenutku registar PC pokazuje na memorijsku ćeliju &1002 u koju je upisana naredba koja bi bila izvršena da tok programa nije nasilno promenjen naredbom bezuslovnog skoka Mikroprocesor će sabrati sadržaj registra PC (&1002) i broj &20 pa će jako dobijenu vrednost staviti u PC Izvršavanje programa će tako biti nastavljeno od instrukcije čija je adresa &1020.

Čemu bi mogla da posluži ova komplikacija? Program koji u celini napišemo korišćenjem isključivo relativnog adresiranja će korektno raditi ma u koji ga segment memorije učitali JR PC+&20 izaziva preskakanje sledećih &20 bajta, bez obzira gde ova instrukcija bila smeštena. Treba međutim da kažemo da je skoro nemoguće a i sasvim nepotrebno pisati program koji radi gde god da ga smestite — daleko je lakše izmeniti ORG i asemblirati program na drugo mesto Relativni skokovi se zato koriste isključivo radi uštede memorijskog prostora kod kratkih skokova koji su inače daleko češći od dugih JP &2020 zauzima tri a JR PC+&20 samo dva bajta Za šta je onda nekome potrebno apsolutno adresiranje? Korišćenjem jednog jedinog bajta za pamćenje rastojanja omogućeni su samo skokovi dugi do 127 bajta (broj iz JR se posmatra kao označen što znači da su mogući i skokovi unazad) dok iza JP možemo da navedemo bilo koju adresu u čitavoj memorijskoj mapi

Većina asemblera tretira relativne skokove kao i apsolutne pisaćete JR DALJE baš kao što biste pisali i JP DALJE Sam asembler će pronaći labelu DALJE i izračunati njeno rastojanje od lokacije instrukcije JR formirajući tako korektan adresni deo ove naredbe Preporučili bismo vam da se ugledate na vaš asembler i na pravite razliku između JP i JR uvek koristite relativne skokove, a apsolutnim pribegnite samo ako vaš asembler prijavi grešku tipa JUMP TOO LONG

Konstruktori Z80 su omogućili i relativne uslove skokova JR C i JR Z, odnosno JR NC i JR NZ. Što se indirektnog adresiranja tiče instrukcija JP (HL) JP (IX) i JP (IY) omogućavaju skok na instrukciju koja je upisana u memorijsku ćeliju čiji je broj upisan u HL, IX odnosno IY LD HL,&3000 JP (HL) je na primer ekvivalentno sa JP &3000

Posebna instrukcija relativnog skoka je DJNZ neka vrsta ekvivalenta FOR NEXT petlje u bejziku Nju ćemo upoznati posmatrajući primer na slici 14

```
3100                 ORG   &3100
3100 21A030          LD    HL,&30A0
3103 3EA0            LD    A,&A0
3105 47              LD    B,A
3106 3E00            LD    A,0
3108 77       PETLJA LD    (HL),A
3109 2B              DEC   HL
310A 10FC            DJNZ  PETLJA
310C C9              RET
```

slika 14

Dalji program upisuje nule u memorijske ćelije čije su adrese &3000 &3001 ... &30A0 U početku upisujemo &30A0 u registar HL koji će uvek pokazivati na memorijsku ćeliju u koju treba upisati nulu U registar B (deo registra BC koji kao što znamo predstavlja Byte Counter, tj brojač bajtova) upisujemo &A0 ukupan broj memorijskih ćelija koje treba izbrisati Zatim ulazimo u petlju

Usvojili da se pre izvršavanja programa u ćeliji 100 nalazio broj &73 odnosno %01110011 novi sadržaj je

```
   01110011
OR 10000000
-----------
   11110011
```

Videvši da smo jednostavno OR setovali sedmi (broji se od nule sećate se?) bit nekog bajta, dovoljno smo sposobni da sastavimo program koji će postaviti sadržaj BB bajta u DDCCBBAA na vrednost koju želimo Pretpostavimo da se ta vrednost nalazi u akumulatoru i to na mestima 2 i 3 (kada upoznamo šiftovanja moći ćemo tu i da je dovedemo!) Da promenimo vrednost tražena dva bita prvo ćemo ih postaviti na nulu (kako bismo uništili eventualnu ranije vrednost) a zatim ih primenom operacije OR postaviti na novu vrednost

LOAD (101),A	Vrednost akumulatora privremeno smeštamo u 101
LOAD A,(100)	Sadržaj ćelije 100 u akumulator
AND A,%11110011	Resetuju se bitovi 2 i 3 (postavljaju na 0)
OR A,(101)	Bitovi se postavljaju u traženo stanje
LD A,(100)	Rezultat u memoriju
HALT	

Neka je staro stanje ćelije 100 bilo %11010101 i neka se u akumulatoru nalazio broj %00001000 koji prema našoj konvenciji treba da promeni stanje ćelije 100 u &11011001 Proverimo da li će se to zaista i dogoditi

```
    11010101
AND 11110011
------------
    11010001

    11010001
OR  00001000
------------
    11011001
```

X	not X
0	1
1	0

slika 5

Osim konjunkcije i disjunkcije konstatujemo i logičku negaciju (NOT) čija je istinitosna tabela date na slici 5 Primena ove operacije je sasvim jednostavna svaka jedinica se pretvara u nulu a svaka nula u jedinicu NOT 11010101 je 00101010 Jedna od primena negacije je promena znaka rekli smo da se *prvi komplement* broja dobija kada se svaka jedinica u tom broju pretvori u nulu a svaka nula u jedinicu sada znamo da se ovakvo pretvaranje naziva *negacijom broja* Ukoliko nam je potreban *drugi* ili *potpuni komplement* broja rezultatu negacije ćemo prosto dodati jedan

```
not (not A) = A          X and (Y or Z) = (X and Y) or (X and Z)
                         X or (Y and Z) = (X or Y) and (X or Z)
A and 1 = A
A and 0 = 0
A or 0 = A               not (X and Y) = not X or not Y
A or 1 = 1               not (X or Y) = not X and not Y
```

slika 6

Na slici 6 je dato nekoliko formula koje povezuju konjunkciju disjunkciju i negaciju Nama naročito potreba da se dublje upuštate u ova formule ni da ih pamtite ali nije nemoguće da će vam neka od njih zatrebati u docnijem redu pa zato treba zapamtiti da ona postoje Ukoliko ste se čitajući ovo zainteresovali za matematičku logiku preporučujemo vam neki srednjoškolski ili univerzitetski udžbenik iz koga ćete saznati mnogo više o Bulovoj algebri tablicama istinitosti valjanim formulama i sličnim zavrzlamama

Pomeranje bitova

Posmatrajmo brojeve 115 230 i 204 i pokušajmo da uočimo neku vezu između njih Na prvi, drugi i treći pogled veze nema Ukoliko imate malo smisla za enigmatiku setićete se da govorimo o logičkim operacijama sa binarnim brojevima pa ćete naše brojeve predstaviti u tom obliku i dobiti %01110011 %11100110 i %11001100 Veza postaje očigledna drugi broj je dobijen kada smo odbacili prvu cifru prvog i dodali mu nulu na kraj treći broj je dobijen kada smo isto to uradili sa drugim Sledeći broj u nizu bi dakle bio %10011000 ili 152.

Tvrdnju *sledeći broj se dobija kada prethodnom dopišemo nulu na kraj i odbacimo njegovu prvu cifru* možemo da iskažemo i na drugi način *sledeći broj se dobija kada sve cifre prethodnog pomerimo ulevo za jedno mesto a na upražnjeno mesto poslednje cifre upišemo nulu* Ovo je ujedno i najjednostavnija definicija šiftovanja ulevo

Šiftovanje udesno je vrlo slična operacija pri čemu bi se od broja %01110011 dobio najpre broj %00111001 a zatim broj %00011100 Šta međutim treba da se dobije šiftovanjem broja %11001000 udesno za jedno mesto? Bez mnogo razmišljanja bismo rekli da je rezultat %01100100 i jako pokazali da smo zaboravili ono o čemu je u prethodnom poglavlju bilo govora

Pretvorimo broj 8 u binarni broj %00001000 i šiftujemo ga za jedno mesto udesno. Dobijemo %00000100 ili dekadno 4. Ako ponovimo operaciju dobićemo dekadno 2 i pri novom ponavljanju jedinicu. Vidimo da se šiftovanje udesno svodi na deljenje broja sa dva, slično tome bismo mogli da pokažemo da se šiftovanjem ulevo broj množi sa dva. Pošto smo ovo razumeli vratimo se prethodnom primeru broj %11001000 pretvoren u dekadni sistem daje 200 dok %01100100 predstavlja broj 100 ili polovinu broja 200. Za sada je sve u redu. Međutim, rekli smo da se u memorijske ćelije često smeštaju **označeni brojevi**, pri čemu se negativni brojevi pamte korišćenjem potpunog komplementa. Obzirom da je vodeći bit broja %11001000 jedinica, radi se o negativnom broju. Dobijamo da je %11001000 = −%00111000 = −56. S druge strane vodeći bit broja %01100100 je nula što znači da se radi o pozitivnom broju 100. Kako polovina od −56 nikako ne može da bude +100, operacija šiftovanja označenog broja je dala pogrešan rezultat!

U čemu smo pogrešili? Samo u tome što smo šiftujući broj udesno na mesto vodećeg (sedmog) bita stavili nulu i tako pokvarili znak rezultata. Ispravno bi bilo da smo na to mesto ponovo upisali bivši vodeći bit čime se znak broja koji taj bit opisuje ne bi promenio. Tako bi se šiftovanjem broja %11001000 udesno dobio broj %11100100 = −%00011100 = −28 što je upravo polovina od −56. Ovakvo šiftovanje kod koga se vodi računa o znaku broja, naziva se *aritmetičkim*, dok se šiftovanja kojim se od %11001000 dobija %01100100 naziva *logičkim*. Mikroprocesori mogu da izvrše bilo koje od ovih šiftovanja, prvo ćete koristiti kada u memorijskim ćelijama čuvate označene brojeve, a kada pišete neku vrstu matematičkog programa, a drugo kada operišete sa bitovima u cilju njihovog pakovanja ili testiranja.

Z80 omogućava relativno složena logička šiftovanja brojeva o kojima ćemo još dosta govoriti. Za sada je važno samo znati da se kod tih šiftovanja vodeći ili krajnji bit ne gubi već se prenosi u jedan od flagova nazvan indikator prenosa. Postoji mogućnost i da se bit iz indikatora prenosa vrati u šiftovani broj, a takođe i da se „izgubljeni" bit vrati na kraj broja; tako bismo od %10010001 dobili %00100011. Ovakva operacija se iz očiglednih razloga naziva rotiranjem ulevo.

Instrukcije i njihovo kodiranje

Kada biste bilo koji od programa koje smo do sada pisali pokušali da „saopštite" mikroprocesoru koji upoznajemo, rezultati bi bili nikakvi. Pre svega za zadavanje instrukcija smo koristili neke reči LOAD, COMPARE AND HALT ali smo prećutkivali način na koji bi one mogle da se unesu u računar. Možda ih otkucate umesto bejzik programa? Pokušajte i pozdravite vas Syntax error ili, pošto ste načuli da se mašinski programi unose primenom instrukcija POKE, da otkucate POKE LOAD? Sličan pozdrav. Da bi računar mogao da izvrši mašinski program on mu mora biti saopšten u prihvatljivom kodiranom obliku!

Mnemoničke skraćenice

Instrukcije se kodiraju u binarne brojeve pri čemu kod svake instrukcije obično zauzima jedan bajt. Osim koda koji bi na primer zamenjivao skraćenicu LOAD računaru treba da saopštimo i adresni deo naše instrukcije (sećate se onoga LOAD *A %11001100?)* za koji se obično koriste sledeći sukcesivni bajtovi memorije. Različite instrukcije zajedno sa svojim adresnim delom mogu da zauzmu jedan, dva ili više bajtova pri čemu ovaj broj varira od instrukcije do instrukcije.

Kada se već instrukcija moraju pretvarati u brojeve, zašto izmišljate reči koje te brojeve zamenjuju? Zato što je mašinski jezik i onako po prirodi težak i nečitljiv pa postojanje naziva instrukcija (tzv. *mnemoničkih skraćenica*) olakšava kako pisanje programa tako i njegovo docnije prepravljanje. Samu transformaciju reči u binarne brojeve je u stanju da obavi kompjuter pomoću programa koji nazivamo *asembler* i kome posvećujemo naše sledeće poglavlje. Mnemoničke skraćenice su stvar konvencija, možemo da se odlučimo da instrukciju čiji je kod &A9 zovemo LD ili LOAD ili samo L, i bilo koja od tih varijanti će rezultirati istim mašinskim programima. Slično je i u bejziku: naredbe PRINT bi sasvim lepo mogla da se zove WRITE, DISPLAY ili TYPE. Konvencija su međutim jednom usvojena i ne možemo da ih menjamo pa se na njih treba privići. Posle određenog broja sati sedenja uz računar naučićete sva bitna skraćenica napamet i kucati ih bez mnogo razmišljanja.

Korišćenje mnemoničkih skraćenica krije u sebi veliku opasnost: tendenciju da se uobrazi postojanje nekih naredbi. Ukoliko na primer znate da postoji naredba JUMP (HL) moguće je da po analogiji upotrebite i naredbu CALL (HL) koja na primer ne postoji! Program koji smo nazvali asemblerom će vas jasno upozoriti na grešku, ali vam to neće mnogo pomoći ako ste čitav segment programa zasnovali na nepostojećoj naredbi. Setovi instrukcija šesnaestobitnih i još moćnijih procesora su daleko bogatiji i njihovi su se konstruktori potrudili da postignu veliku simetričnost: svaka instrukcija ima sve adresne modalitete i instrukcije slične prirode su konsistentne (ovakav set instrukcija nazivamo *ortogonalnim)*. Osmobitni mikroprocesori, međutim, zbog uštede memorije i jednostavnosti konstrukcije imaju razne modalitete adresiranja koji su pristupačni samo kod nekih instrukcija i za njihovo korišćenje je potrebno bar u početku, neprestano konsultovati tabelu naredbi koja je data u našem prethodnom umetku. Korišćenje je tabela ćete upoznavati kako ovaj tekst bude napredovao; za sada samo bacite jedan pogled na nju i ukoliko za to imate uslova napravite par fotokopija — jer će fotokopija postati redovan gost na vašem pisaćem stolu. Videćete da su u tabelama pobrojane skraćenice instrukcija, kratak opis koje se zove SRA i čije je dejstvo prikazano u tabeli instrukcija. Postoje i više nego čudne naredbe RLD i RRD koje razmenjuju polovinu bitova akumulatora sa memorijom i koriste se za rad sa BCD brojevima o kojima, kao što rekosmo, ukratko govorimo tek u budućim napisima.

Bezuslovni i uslovni skokovi

Kada bejzik interpretator vašeg računara izvrši neku instrukciju on automatski prelazi na izvršavanje sledeće i tako dalje — sve dok ne naiđe na neko STOP ili END. Korišćenjem GOTO naredbe se naravno može zahtevati da se ovaj tok izvršavanja izmeni, dok se IF ... THEN ... ELSE konstrukcijom postiže izvršavanje sledeće naredbe samo ako su neki uslovi ispunjeni. Bez ovakvih naredbi ne bismo mogli da sastavimo čak ni najjednostavnija programa, što znači da nešto odgovarajuće mora da bude raspoloživo i na mašinskom jeziku.

Bezuslovni skokovi

Najjednostavnija naredba bezuslovnog skoka je JP — potpuni ekvivalent GOTO. Iza JP se nalazi adresa memorijske ćelije u koju je upisana naredba koju sledeću treba izvršavati, što znači da JP &3000 jasno podrazumeva da se instrukcija na &3000 zaista može izvršiti, kako mnoga instrukcija zauzimaju dva, tri ili čak četiri bajta može da se dogodi da se u memorijskoj ćeliji &3000 ne nalazi početak neke instrukcija već njen drugi ili treći bajt. Dok bejzik interpretator u konfliktnim situacijama prijavljuje grešku, mikroprocesor ne poseduje apsolutno nikakav način da prosudi da li se na &3000 nalazi instrukcija koju ste vi planirali za izvršavanje, on će pokušati da izvršava ono što tamo nađe što će dovesti do nepredvidljivog toka događaja i moguće potpunog blokiranja računara.

Iako Z80 razume jedino instrukcije tipa JP &3000, za nas je ovakvo korišćenja apsolutnog adresiranja sasvim nepogodno. Pre svega retko su nam potrebni skokovi na ovako „okrugle" lokacija — obično programiramo prelazak na neku instrukciju u programu. Pri pisanju tog programa nam je jasno poznata njegova početna adresa u memoriji (pišemo ja iza ORG) ali nam nija poznata početna adresa svake njegove instrukcija, da bismo je našli moramo da izmčunamo koliko bajta memorije zauzima svaka njoj prethodna instrukcija pa da tako dobijena brojeve saberemo i dodamo na vrednost ORG. Takvo računanje i sabiranje je neprijatan posao pošto je greškama koji, da stvar bude još gora, moramo da ponavljamo kada god u programu napravimo bilo kakvu izmenu koja će produžiti ili skratiti neki njegov segment. Zbog toga svi bolji asembleri omogućavaju rad sa *labelama*.

Reč labela („label") može da se prevede kao „oznaka" — radi se o nekom simbolu kojim označavamo određeni segment programa. Asembleri obično omogućavaju da ime labele bude bilo koja reč od 3—6 slova sa izuzetkom mnemoničkih skraćenica naredbi; ako iskoristimo labelu JP računar je neće razlikovati od instrukcija JumP! Jasno je da ćemo ime labela birati tako da asocira na svrhu segmenta programa koji označavamo; česta imena labela su RADNI, RAČUN, DALJE, GREŠKA, KRAJ i slično.

```
3000                    ORG     &3000
3000 210000             LD      HL 0
3003 7E        CIKLUS   LD      A (HL)
3004 77                 LD      (HL) A
3005 23                 INC     HL
3006 C30330             JP      CIKLUS
```

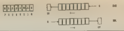
slika 12

Na slici 12 je prikazan program koji koristi naredbu JP da bi realizovao beskonačnu petlju; ovaj program upisuje u svaku memorijsku ćeliju njen prethodni sadržaj i tako u beskonačnost! Labela CIKLUS označava početak petlje, pa je JP CIKLUS ekvivalentno sa JP &3000 s tim što je JP CIKLUS daleko univerzalnija: jednostavnom promenom naredbe ORG ovaj program možemo da asembliramo u bilo koji segment memorije. Ukoliko se odlučite da otkucate i isprobate ovaj program uzmite u obzir da će on eventualno izazvati krah sistema ukoliko vaš računar (što inače nije uobičajeno za Z80) koristi memorijski mapirana periferijska uređaje i da ćete moći da ga prekinete jedino pritiskom na RESET taster.

Labela osim za skokove možete da koristite i kao zamenu za brojeve. Na početku programa možete na primer da napišete RAMTOP EQU &5CB2 pa će docnija LD (RAMTOP) HL biti ekvivalentno sa LD (&5CB2) HL. U ovom slučaju korišćenja labela pomaže jedino boljoj čitljivosti programa jer ćete kada ga docnije budete gledali zaboraviti da je &5CB2 adresa sistemske promenljive RAMTOP pa ćete morati da konsultujete razne tabele da biste razumeli segment programa koji ste sami napisali.

Uslovni skokovi

Naredba JumP izaziva tzv. bezuslovni skok — naredba čija je adresa navedena iza JP se izvršava u svakom slučaju. U praksi ćemo daleko češće koristiti takozvane uslovne skokove koji su omogućeni postojanjem flagova. Naredba JP Z JEDNAK će na primer izazvati skok na segment programa označen labelom JEDNAK **samo ako** je Z(ero) flag setovan. Ukoliko želimo da se skok na labelu izvrši ukoliko je indikator nule resetovan koristićemo JP NZ JEDNAK (Jump If Not Zero).



SLIKA 9

budemo želeli da saviano od ulaznih veličina programa setujemo neki bit morećemo da pripremimo odgovarajuću masku pa da koristimo OR

Ako imamo naredbu za setovanje nakog bita, verovatno postoji i naredba za njegovo resetovanje Njen opšti oblik je RES b t gde se o b i o t ne može reći ništa posebno novo Ova naredba se kao što smo videli u uvodnim poglavljima može simulirati uz pomoć AND

Testiranja bitova se obavlja primenom naredba BIT b t Nju bismo mogli da prevedemo na srpskohrvatski kad *da li je bit broj b odredišta t setovan?* Ako je odgovor na ovo pitanje potvrdan (bit je setovan) zero fleg će biti resetovan a ako je odgovor odrečan (flag je resetovan) zero fleg će biti setovan Ovakva konvencija izglade naopako ali šta da se radi Ukoliko da rezimiramo želimo da nastavimo program od mesta koje je označeno sa DALJE samo ukoliko je sedmi bit akumulatora setovan izvršićemo

BIT 7,A	testiraj bit 7 akumulatora.
JP NZ, DALJE	ukoliko zero fleg **nije** setovan
	bit 7 **je** setovan pa skoči na DALJE

Šiftovanje i rotiranje

Govoreći o logičkim operacijama u uvodnim poglavljima ukratko smo obradili šiftovanje i pomenuli razliku između logičkog i aritmetičkog šiftovanja i rotiranja U tom vam je trenutku čitava stvar mogla izgledati komplikovano ali je ona prava šala prema različitim vrstama šiftovanje i rotiranja koja Z80 podržava. Jasno je da svi tipovi šiftovanja nisu neophodni oni su pripremljeni da vam olakšaju život Dok ste međutim početnik mnogo različitih operacija vam umesto toga zagorčava život pa predlažemo da za početak upoznate samo najosnovnije Kada vam neka od specijalnih operacija ovoga tipa bude zatrebala tako ćete je razumeti

Logičko šiftovanje ulevo i udesno je prikazano na slici 10 Vidimo da se SLA i odnosno SRL t pomeramo bitove odredišta t na levo odnosno na desno za jedno mesto umesto nultog (za SLA) odnosno sedmog bita (kod SRL) u odredište ulazi nula dok bit koji ispadne usred pomeranja odlazi u indikator prenosa Raniji sadržaj indikatora prenosa se jednostavno gubi

Vidimo da smo primenom instrukcija SLA odnosno SRL izgubili bivše stanje indikatora



slika 11

prenosa a u odredišta upisali jednu nulu Zašto da na izbegnemo gubitak tog bita i veštačko generisanje nule prepisavši prethodno CY na kraj odnosno početak odredišta? Tako dobijamo operacije RL t i RR t rotaciju odredišta t i indikatora prenosa za jedno mesto ulevo odnosno udesno Obe ove operacije su prikazane na slici 11

SLA SRL RL i RR su osnovna šiftovanja i rotacije ostatak ovoga poglavlja slopodno možete da zanemarite Uz određenu dozu hrabrosti možete naravno i da nastavite sa čitanjem

Šta da radimo ako nam je potrebno da prosto rotiramo sadržaj nakog registra za jedno mesto ulevo? Možemo da izvršimo SCF zatim RL t a da onda ako je indikator resetovan izvršimo RES 0 t Da vidimo kako to radi ako se u registru B nalazio broj %01010011 Posle SCF ovaj broj se nije promenio ali je indikator prenosa setovan Posle RL B vodeći bit broja (0) odlazi u CY petali bitovi se pomeraju za po jedno mesto ulevo a bivše stanje CY flega (1) dolazi na kraj broja dobili smo %10100111 Pošto je indikator prenosa setovan izvršava se RES 0 B pa %10100111 postaje %10100110 baš ono što smo želeli Isti rezultat može međutim da se dobije i primenom naredba RLC t (*rotate left circular* ili u slobodnom prevodu uz malo pleonazama *kružno rotiraj*). Dejstvo ove i odgovarajuće naredbe RRC t koje rotira odredište udesno je prikazano u tabeli instrukcija

Kod šesnaestobitnih load instrukcija smo videli da su konstruktori Z80 ponekad bili raspoloženi da nam omoguće štednju bajtova u programu pripremajući specijalna slučajeve nakih instrukcija postoji na primer naredba LD rp (nnmm) gde je rp bilo koji registarski par (pa i HL) a onda i instrukcija LD HL (nnmm) koje zauzima bajt manje Sve instrukcija za rotiranje i šiftovanja koje smo do sada upoznali rade sa bilo kojim registrom memorijskom ćelijom na koju pokazuje HL ili neki od indeks registara pa zauzimaju po dve tri ili četiri bajta programa Šiftovanja i rotiranja se međutim obično izvode nad sadržajem akumulatora pa postoje jednobajtne naredba RLCA RLA RRCA i RRA kao specijalni slučajevi RLC RL RRC i RR Ukoliko dakle vašem asembleru kažete RR A potrošićete dve bajta dok ćete ako zaboravite blanko pa napišete RRA potrošiti samo jedan Pa neka posle neko kaže da se ne isplati grešiti

Z80 je opremljen naredbom za aritmetička šiftovanja udesno (deljenja označenog broja sa 2)

njihove funkcije načini adresiranja koji ih prate kodovi i podatak koji vas za sada neće interesovati broj mašinskih ciklusa koji su potrebni da se instrukcija izvrši Na osnovu ovog broja možete da izračunate koliko se vremena izvršava neka instrukcija ako vaš računar radi na frekvenciji od 2 MHz, jedna mašinski ciklus traje 1/(2MHz)=0.5 mikrosekundi Kada jednom budete programirali neki kontroler koji radi u realnom vremenu (npr crta sliku na ekranu televizora) videćete da će vas vreme izvršavanja instrukcija naglo zainteresovati

Memorija i adresiranje

Memoriju možete da zamislite kao niz ćelija sa slika 7 Svaka ćelija ima svoju adresu koja se kreće između 0 i 65535 (ili ako više volite heksadekadna brojeve između &0 i &FFFF) Svaka ćelija se sastoji od osam bitova koji su obeleženi brojevima 0—7

65536 bajta iznosi 65536/1024=64 kilobajta i to je najveća memorija kojom može da bude opremljen računar sa Z80 6502 ili bilo kojim drugim osmobitnim mikroprocesorom (kod 16-bitnih mikroprocesora broj se penje do 16 Mb) Ovaj *adresni prostor* mora da buda podeljen na ROM u koji će biti smešten operativni sistem računara i interpreter (i) programskih jezika kao što je bejzik i RAM u koji će biti upisivani programi koje korisnik piše. Za sada ćemo reći da se ROM kod računara sa Z80 počinje od adrese &0000

Većina instrukcija koje zadajemo mikroprocesoru se bavi manipulacijom (prenošenjem sabiranjem testiranjem) brojeva koji su smešteni u neka memorijska ćelija što znači da se uz kod instrukcije mora navesti i neka oznaka na osnovu koje će računar zaključiti gde se nalaze podaci koje treba da obrađuje Ta oznaka se zove *adresni deo instrukcije* U zavisnosti od načina na koji mikroprocesor transformiše adresni deo instrukcije u podatak sa kojim treba manipulisati razlikujemo razne *modove adresiranja* od kojih ćemo neke upravo upoznati.

Implicitno adresiranje

Najjednostavniji slučaj adresiranja je izostavljanje adrese — adresa je tada implicirana smislom naredba pa se ovakvo adresiranje naziva *impliciranim (implied)* Posmatrajmo na primer naredbu SCF čiji je jedini smisao da setuje (postavi na jedinicu) indikator prenosa koji smo već pominjali iz koda ove naredba se zna sve što je potrebno za njeno izvršavanje pa bi bilo kakva adresa (npr SSF carry) bila suvišna

Registarsko adresiranje

Slično impliciranom je *registarsko adresiranje* koje podrazumeva da se argument nalazi u nekom od registara mikroprocesora. Instrukcija INC A na primer, dodaje jedan sadržaju akumulatora.

Neposredno adresiranje

Malo je složeniji slučaj instrukcija LD A,25 koje u akumulator dovodi broj 25 Broj kojim treba manipulisati se ovde nalazi u okviru same instrukcije pa se tako adresiranje naziva *neposrednim (immediate)* Instrukcije ovoga tipa ćemo koristiti kada želimo da dovodimo konstante u memoriju smeštamo ih u registre ili na neki drugi način manipulišemo njima.

Apsolutno adresiranje

Došlo je najzad vreme da počnemo da radimo sa sadržajima memorijskih ćelije i upoznamo *apsolutno adresiranje* posmatraćemo instrukciju LD A (&1000) koja u akumulator dovodi osam bita ćelije čije je adresa &1000 Ova naredba je u ovakvom obliku sasvim jasna No šta da kažemo na LD HL (&1000)? Iako još nismo govorili o šesnaestobitnim registrima iz oznaka ove instrukcija može da se pretpostavi da je HL naziv registra koji je nastao slapijivanjem registara H i L i koji tako ima 16 umesto 8 bita Memorijska ćelija &1000 ima, kao što znamo samo osam bita što znači da bi pri doslovnom shvatanju navedena naredba polovina registra HL ostala prazna Konstruktori Z80 su odlučili da se ova instrukcija izvrši tako što se u HL smeste sadržaji memorijskih ćelija &1000 i &1001 i to tako da memorijska ćelija &1000 ode u registar L a memorijska ćelija &1001 u registar H Ako su na primer ćelije &1000 i &1001 sadržavale brojeve &20 i &3A u registar HL će stići broj &3A20 primetimo da ćelija sa nižom adresom čuva manje značajan a ćelija sa višom adresom više značajan bajt šesnaestobitnog broja (znate li šta su više i manje značajna cifra? U broju 1234 cifra 1 označava hiljada i kao takva je daleko značajnije od cifra 4 koja označava jedinice ako pri pisanju ovoga broja napišete 2 umesto 1 pogrešićete za 81% dok ćete ako zamenite 4 sa 5 pogrešiti samo za 0.08%)) Ovo je konvencija koja je usvojena praktično kod svih procesora kako osmobitnih tako i moćnijih

Relativno adresiranje

Ako postoji *apsolutno adresiranje* verovatno postoji i *relativno!* Upravo tako kod apsolutnog adresiranja specificiramo memorijske ćelije navodeći njihovo tačno mesto u memorijskoj mapi dok kod relativnog adresiranja navedimo rastojanja podatka od instrukcija koja se izvršava Prototip ovakve instrukcije bi bilo JR PC + &20 PC je skraćenica od *Program Counter* to je ime

registra koji prati izvršavanje programa i stalno sadrži adresu instrukcije koja će se sledeća izvršavati Ta adresa će biti sabrana sa konstantom &20 a zatim će biti izvršen skok na naredbu čiju smo ovim sabiranjem dobili Za šta bi mogla da se iskoristi ovakva komplikacija? Ukoliko napišemo čitav program koristeći ovakvo adresiranje za sve naredbe (posebno naredbe skoka) on će se korektno izvršavati bez obzira na to u koji ga segment memorije upišemo Kako ovakva mogućnost nije naročito korisna relativno adresiranje je kod jednostavnih procesora kao što je Z80 zastupljeno samo kod malog broja instrukcija Suprotno toma kod velikih kompjuterskih sistema koji istovremeno opslužuje više korisnika relativno adresiranje predstavlja osnovni modalitet rada

Indirektno adresiranje

Indirektno adresiranje je daleko značajnije bez njega se teško mogu ostvariti iole složenije programaraka samisli Verujući da je većina čitalaca ovog umetka dobro upoznata sa bejzikom objasnićemo indirektno adresiranje na primeru naredbi PRINT A(100) i PRINT A(I) Obe naredbe treba da prikažu na ekranu sadržaj nekog elementa niza A Kod prve (PRINT A(100) je broj tog elementa fiksiran u okviru naredbe pa ona predstavlja primer *apsolutnog* adresiranja Naredba PRINT A(I) sa druge strane naređuje bejzik interpretatoru da najpre proveri vrednost promenljive I **a onda** da ispiše vrednost elementa niza čiji broj ona čuva To je *indirektno* adresiranje jer se u okviru naredbe ne navodi pozicija argumenta već mesto na koma može da se nađe ta pozicija U mašinskom programiranju Z80 se umesto promenljiva za indirektno adresiranje koriste registri instrukcija LD A (HL) će ukoliko se u registru HL nalazi broj &1234 dovesti u akumulator sadržaj memorijske ćelije čija je adresa &1234

Indeksno adresiranje

O *indeksiranom adresiranju ćemo* govoriti docnije obzirom da je ono nešto komplikovanije od svih do sada pobrojanih Ukratko u okviru same instrukcije ili u nekom registru se nalazi tzv *bazna adresa* a na nekom drugom mestu tzv *ofset*. Računar sabira baznu adresu i ofset, a zatim pristupa ćeliji čija je adresa određena ovim zbirom Ako se na primer u registru IX nalazio broj &2000 naredba LD A (IX+&20) će u akumulator dovesti sadržaj ćelije čija je adresa &2000+&20=&2020 Indeksirano adresiranje je izuzetno značajno za većinu mikroprocesora, ali ga Z80 koristi u gotovo zanemarljivoj meri

Asembleri

Ulogu asemblera (ili asembler programa) već poznajemo oni treba da prevedu program pisan mnemoničkim skraćenicama (zvaćemo ga *izvorni program*) u *objektni kod* — pravi mašinski program koji će biti upisan negde u memoriju Ukoliko nameravate da vam ovaj umetak posluži za bilo šta osim za ukras ***nabavite asembler program za vaš kompjuter*** To nije naročito velika investicija (ovakvi programi su kod preprodavaca jeftini valjda zato što slabo koga interesuju) a predstavlja *neophodan* uslov za red na mašinskom jeziku Pretpostavljajući da asembler program (ili kraće asembler) već posedujete nećemo se baviti prevođenjem instrukcija na pravi mašinski jezik ukoliko dođete do toga da vam nešto takvo zatreba verovatno ćete biti dovoljno dobar poznavalac mikroprocesora da se sami snađete Ovo poglavlje je zato posvećeno isključivo upotrebi asemblera

Asemblerske direktive

Asembler je program kao i svaki drugi pravi se u mnogo verzija i svaka od njih treba da ima svoja karakteristika i svoja uputstvo za upotrebu U Računarima 12 i Računarima 13 smo opširno opisali upotrebu najpoznatijeg asemblera za spektrum i amstrad koji se zove DEVPAC dok smo se asemblerom za galaksiju bavili u okviru umetka ROM 2 iz Računara 3 ukoliko što nije naročito verovatno koristite neki drugi program ili drugi računar moraćete da pročitate originalno uputstvo što vam u svakom slučaju savetujemo i kada se radi o DEVPAC u Videćete da se asembler sastoji iz *editora* koji omogućava da unosite ispravljate i snimate program na traku ili disketu i *prevodioca* koji ovako formiran tekst (kažemo tekst jer su mnemonička skraćenice obična reči) prevodi u objektni kod i upisuje ga u memoriju Editori se razlikuju od asemblera do asemblera dok prevodioci rade manje više jednako šte znači da ćemo im posvetiti posebnu pažnju

Da bi prevodilac mogao da smesti program u memoriju treba mu saopštiti adresu od koje program počinje Ta adresa se naziva *origin* i obeležava sa ORG Ukoliko se na početku programa pisanom mnemoničkim skraćenicama nađe ORG &3000 rezultujući mašinski kod će biti upisan u memoriju počevši od adrese &3000

Reč ORG nije mašinska instrukcija pošto se ona ne prevodi u binarni broj koji bi bio smešten u memoriju Ona umesto toga daje prevodiocu uputstvo o tome šta da radi sa pravim instrukcijama pa se naziva *asemblerskom direktivom* Asemblerskih direktiva jasno ima mnogo pa ćemo ih

A	Y	[illegible]
0	0	0
0	1	1
1	0	1
1	1	0

slika 8

Da vidimo za šta može da se iskoristi ova ekskluzivna operacija XOR A zvuči prilično glupo računamo A XOR A i rezultat opet šaljemo u A! Međutim primetimo da se na taj način vrši ekskluzivno ili (tzv *ili/ili*) nad jednakim bitovima Kako je 1 XOR 1=0 i 0 XOR 0=0 u akumulatoru će se naći sve same nule kao da smo napisali LD A 0! Mogli smo naravno da napišemo i ovo ali bismo tada utrošili jedan bajt više i dobili program koji se sporije izvršava

Došlo je vreme da objasnimo i smisao naredbe AND A koja briše sadržaj indikatora prenosa Briše ga zato što u tabeli instrukcija piše da svaka od naredbi AND OR i XOR stavlja nulu u CY Treba da dodamo da će vrednost akumulatora ostati nepromenjena jer će se svaki bit logički množiti sa samim sobom jedinica ostaje jedinica jer je 1 AND 1 = 1 a nula ostaje nula jer je 0 OR 0=0 Mogli smo sasvim ravnopravno da upotrebimo i OR A, dok bi XOR A takođe obrisalo indikator ali bi uzgred obrisalo i akumulator!

Još jedna lepa osobina naredbi AND OR i XOR je da postavljaju indikator nula (Z) ukoliko se u akumulatoru dobija nula Ako nam je na primer potrebno da ispitamo da li je u akumulatoru nula napisaćemo AND A a zatim JP Z NULA *(=Jump if Zero set,* skoči ako je zero flag setovan) Sadržaj akumulatora ako je različit od nula neće biti izgubljen jer smo već rekli da AND A utiče samo na flegove

Šesnaestobitna aritmetika

Sve osmobitne aritmetička operacije se obavljaju nad sadržajem akumulatora, dok ulogu šesnaestobitnog akumulatora igrava registar HL Druga razlika između osmobitnih i šesnaestobitnih aritmetičkih naredbi je što ove druge utiču na mali broj flegova — DEC A će, na primer setovati Z(ero) flag ako je sadržaj akumulatora postao jednak nuli, dok će posle DEC HL stanja svih flegova ostati nepromenjena No pođimo redom

Naredbe ADD i ADC imaju isto dejstvo kao i njihovi osmobitni ekvivalenti, ali se primećuje razlika u mnemonici piše se ADD B ali i ADD **HL**,BC a ne kao što bi se dalo pomisliti ADD BC. verovatno je Zilog želeo da se obezbedi od grešaka koja bi mogla da nastanu kada programer izostavi jedno slovo pa se ADD BC pretvori u ADD B Bilo kao bilo opšti oblik ADD i ADC instrukcija je ADD (ADC) HL rp gde je **rp** bilo koji od registarskih parova BC DE HL i SP Za šesnaestobitno oduzimanje se koristi SBC HL rp dok naredba SUB se **ne postoji** ako vam je potrebno obično šesnaestobitno oduzimanje moraćete da resetujete indikator prenosa pa da izvršite SBC Što se indeksnih registara tiče na raspolaganju su vam naredbe ADD IX pp i ADD IY rr pri čemu se na sadržaj IX (IY) može dodavati BC DE SP i IX (IY) **Ne postoje** naredbe ADD IX IY ili ADD IY IX a ni naredba koju ste svakako očekivali ADD IX HL

Naredbe INC se omogućava povećanje a DEC se umanjenje bilo kog registarskog para za jedan pri čemu se stanje flegova ne menja ako se DEC smanjujete brojač u petlji moraćete da dodate nekoliko naredbi koja će proveriti da li je on stigao do nula Što se indeks registara tiče. poslužiće vam naredbe INC IX i INC IY kao i odgovarajuće DEC instrukcije

Operacije sa bitovima

Premda operiše sa bajtovima Z80 je opremljen sa nekoliko veoma moćnih naredbi za red sa njihovim sastavnim delićima — bitovima Osim šiftovanja i rotiranja koja omogućavaju i svi drugi osmobitni mikroprocesori Z80 omogućava i setovanja resetovanja i testiranje bilo kog bita u registrima ili pod određenim uslovima u memoriji

Postavljanje, brisanje, i testiranje

Iako smo nekoliko puta pominjali termine *nulti bit* ili *najznačajniji bit nekog bajta* pogled na sliku 9 neće škoditi Vidimo broj 123 ili &7B koji je upisan u neki registar; najniži ili nulti bit ovoga broja je kao što vidimo jedinica a najviši sedmi bit je nula Ostali bitovi su numerisani brojevima 1—6

Pretpostavimo da je broj &7B upisan u registar B i da Z80 izvrši naredbu SET 7 B Obzirom da već dosta dugo gledamo razne instrukcije dejstvo ova neće biti teško pogoditi ona setuje sedmi (poslednji) bit registra B pa broj %01111011 postaje %11111011=&FB Da li smo ovaj rezultat mogli da dobijemo još nekako? Naravno da jesmo naredbama LD A B OR &80 LD B A možemo da simuliramo SET 7 B uz utrošak četiri umesto dva bajta programa Možda je značajnije što će nam SET 7 B kada docnije budemo posmatrali program olakšati njegovo razumevanje

Naredbom SET ne možemo na žalost da zamenimo OR njen opšti oblik je SET b t gde je *b* broj između 0 i 7 a t, kao i ranije bilo koji od registara (SET b reg) memorijska ćelija na koju pokazuje HL (SET b (HL)) ili neki od indeks registara (SET b (IX + d) odnosno SET b (IY + d)) b je da ponovimo **broj između 0 i 7** to jest broj k ji mera da se zna u vreme pisanja programa ako

Naredbe ADD i ADC utiču i na Z, P/V i S flegove ukoliko je na primer rezultat sabiranja 0 biće setovan Z(ero) fleg. Kako može da se dobija nula sabiranjem dva pozitivna broja? Pokušajte, na primer da saberete &F0 i &10 i dobićete &100 kako u akumulator može da stane svega osam bita rezultata biće upisano 00 i setovan indikator nula. P/V fleg označava nekorektan rezultat sabiranja označenih brojeva o kome smo govorili u drugom poglavlju. Konstruktori Z80 su nas opremili naredbama za testiranje P/V flega, tako da u slučaju ovakve greške možemo da programiramo skok koji će prekinuti izvršavanje programa ili izdati neku poruku.

Pošto smo na jedvite jade shvatili kako se koriste naredbe ADD i ADC treba da kažemo šta mogu da budu njihovi argumenti. Jedan od sabiraka je uvek u akumulatoru u koji se smešta i rezultat. Iza ADD (ADC) navodimo adresu drugog sabirka koji može da bude konstanta (ADD nn npr. ADD &20) da se nalazi u nekom drugom osmobitnom registru (ADD reg) ili da bude u memoriji tako da na njega pokazuje registar HL (ADD (HL)) ili neki od indeksnih registara (ADD (IX+dd) odnosno ADD (IY+dd)). Verujemo da nema potrebe da ponovo objašnjavamo suštinu i primenu neposrednog, apsolutnog, indirektnog i indeksnog adresiranja koja se koristi kod ovih naredbi.

Oduzimanje osmobitnih brojeva

Oduzimanje je sasvim slično sabiranju. SUB s će od akumulatora oduzeti sadržaj s pri čemu s može da bude broj (SUB nn), registar (SUB reg), memorijska ćelija na koju pokazuje HL (SUB (HL)) ili memorijska ćelija adresirana pomoću indeks registra (SUB (IX+dd) odnosno SUB (IY+dd)). SBC s je slična naredba koja ima iste adresne modove ali se bitno razlikuje od SUB po tome što se u akumulator ne stavlja (A-s) nego (A-s-CY) gde je CY kao i ranije stanje indikatora prenosa. Primenom bilo koje od ovih instrukcija indikator prenosa će biti setovan ako je potrebna „pozajmica“, tj. ako smo od manjeg broja oduzimali veći, dok će biti resetovan ako je operacija korektno izvršena. Na taj način prostom zamenom ADD sa SUB i ADC sa SBC u prethodnom primeru možemo da oduzimamo šesnaestobitna brojeva. I kod oduzimanja Z fleg biva setovan ako je rezultat u akumulatoru nula dok se P/V setuje ako je nastupila greška. S(ign) fleg označava znak rezultata koji ima smisla samo ako operišemo sa označenim osmobitnim brojevima.

Specijalan slučaj sabiranja i oduzimanja su instrukcije INC t odnosno DEC t gde t može da bude bilo koji od registara (reg) ili memorijska ćelija na koju pokazuje HL (DEC (HL)) ili neki od indeks registara (INC (IX+dd)) odnosno DEC (IY+dd)). Instrukcija INC (=increment, povećaj) će povećati sadržaj t za jedan dok će ga DEC t za isto toliko umanjiti. Videćemo da su ove naredbe vrlo značajne u petljama gde je potrebna česta promena sadržaja raznih brojača. INC i DEC ne deluju na indikator prenosa što znači da će INC A dati nulu ako se u akumulatoru nalazilo &FF ali se zato Z i S fleg postavljaju u slučaju da je rezultat nula, odnosno da je negativan.

Obzirom da smo pomenuli naredbu SCF koja setuje indikator prenosa pomenimo i CCF *(change carry flag)* instrukciju koja menja njegovo stanje: ako je CY bilo jedan postaće nula, a ako je bilo nula postaće jedan. Brisanje CY možemo umesto sa AND A da izvedemo sa SCF CCF, ali tako trošimo jedan bajt više. U osmobitne aritmetičke instrukcije možemo da ubrojimo i NEG, jednostavnu naredbu koja nalazi drugi komplement sadržaja akumulatora, tj. menja znak broja u akumulatoru. Znajući da je A−B = A+(−B) mogli bismo da zamenimo SUB B sa:

```
PUSH   AF     čuvanje sadržaja akumulatora na steku
LD     A B
NEG
LD     B A    promena znaka broja u B
POP    AF     restauracija sadržaja akumulatora.
ADD    B      A=A+(−B)=A−B
```

U tabeli ćete pronaći i naredbu DAA *(decimal adjust accumulator)* koja se koristi za rad sa BCD *(binary coded decimal)* brojevima. Kako se u ovom umetku bavimo samo osnovnim metodama predstavljanja brojeva, ovoj naredbi posvećujemo tekst u jednom od sledećih brojeva „Računara“.

Logičke operacije

Došlo je vreme da se pozabavimo logičkim operacijama AND, OR, XOR i CPL. Prve dve smo upoznali na početku ove knjige, a poznajemo i poslednju samo pod drugim imenom. CPL je isto što i NOT, naredba koja nalazi prvi komplement akumulatora to jest svaku jedinicu u njemu pretvara u nulu a svaku nulu u jedinicu. AND, OR i XOR su sa druge strane binarne operacije koje imaju po dva argumenta pa ćemo im posvetiti malo više pažnje.

Jedan od argumenata bilo koje od ovih operacija se nalazi u akumulatoru u koji se smešta i rezultat. Drugi može da se nalazi u okviru same instrukcije (AND nn, neposredno adresiranje npr. AND &F0) u nekom od registara (AND reg) ili u memoriji (AND (HL) odnosno AND (IX+dd)), možemo sve u svemu da kažemo da postoje instrukcije AND s, OR s i XOR s gde smo smisao slova s već ranije objasnili.

Nije teško pogoditi šta rade naredbe AND s i OR s: sadržaj akumulatora se bit po bit poredi sa s pa se formira rezultat prema tabelama sa slike 3 i 4. XOR s radi nešto sasvim slično pri čemu se nad bitovima izvodi ekskluzivna disjunkcija koja je definisana tabelom sa slike 5. Ekskluzivna disjunkcija, kao što joj i ima kaže, liči na disjunkciju (OR) ali se od nje malo i razlikuje: 1 OR 1 = 1 ali je 1 XOR 1 = 0!

upoznavati kako nam budu trebale. Neke od njih imaju sasvim banalno i sa aspekta posla koji treba da se obavi nebitno značenje: da li će se pri asembliranju listing prikazivati na ekranu ili će se ispisivati na štampaču, da li će računar prijavljivati sve greške ili samo one fatalne i slično. Neke druge direktive (kao što je ORG koju smo upravo upoznali) su fundamentalno važne.

Pseudoinstrukcije

Osim asemblerskih direktiva i mnemoničkih skraćenica u asemblerskim listinzima mogu da se nađu i *pseudoinstrukcije*. Pseudoinstrukcije, iako ne predstavljaju nijednu od naredbi koju bi mikroprocesor razumeo, u rezultujući kod upisuju određeni sadržaj. Pretpostavimo na primer da naš program treba da ispiše tekst „ZDRAVO“ na ekranu. Ispisivanje se, uzećemo najjednostavniji slučaj, obavlja tako što se poruka ZDRAVO slovo po slovo prepisuje u video memoriju, što znači da slova Z, D, R, A, V i O moraju da budu upisana negde u RAM. Program koji rešava problem bi mogao da izgleda otprilike ovako:

```
LD      HL PORUKA   u registar HL se upisuje adresa početka poruke
LD      DE VIDEO    u registar DE se upisuje adresa početka
                    video memorije
petlja
LD      A,(HL)      slovo poruke u akumulator
CP      0         . kraj poruke je nula bajt.
JP      Z kraj      ako je u akumulatoru 0 ide se na deo
                  . programa označen sa KRAJ
LD      (DE),A      slovo se ispisuje na ekranu.
INC     HL          X se povećava za 1
INC     DE          Y se povećava za 1
JP      petlja    . ispisuje se sledeće slovo
kraj
HALT                kraj rada
poruke
DEFM    ,ZDRAVO
DEFB    0
```

Nije previše važno da shvatite kako tačno radi ovaj program: u globalu posmatrano računar čita slovo po slovo poruke i ispisuje ga na ekranu. Obzirom da bi ispisivanje bilo „mrtva petlja“ na kraj poruke je upisan nula bajt koga program detektuje (naredba COMPARE, CP) i prekida sa radom.

Obratimo pažnju na istaknute pseudonaredbe DEFM i DEFB. Prva od njih označava da prevodilac treba u memoriju da upiše tekst ZDRAVO, svako slovo u po jednu memorijsku ćeliju koristeći naravno ASCII kod. Slično tome pseudonaredba DEFB zahteva od prevodioca da u memoriju upiše bajt 00. Ukoliko nekom prijatelju docnije budete dali mašinski program poput ovoga on će perfektno raditi. Međutim ako taj prijatelj bude pokušao da shvati kako je program napisan, tj. da ga pretvori u mnemoničke skraćenice, naći će se u velikom čudu kada bude pokušao da prevede (odnosno, kako se to stručno kaže, *disasemblira*) deo programa koji predstavlja poruku ZDRAVO. Naći će naime na bajtove od kojih neki predstavljaju instrukcije dok su drugi nedefinisani. Čak i oni koji predstavljaju instrukcije će imati besmislene adresne delove i uopšte neće ličiti na program. U sličnoj situaciji bi se nalazio i sam mikroprocesor da smo zaboravili na naredbu HALT koja prekida njegov rad: pokušao bi da izvrši tekst ZDRAVO kao mašinski program i pri tom verovatno upao u neku vrstu beskonačne petlje.

Disasembleri

Iz čitave ove priče može da se izvede jedan izuzetno važan zaključak: od izvornog teksta programa se jednoznačno dobija objektni kod uz pomoć programa koji nazivamo asembler. Obrnuti proces, dobijanje mnemonički pisanog programa iz objektnog koda (disasembliranje) nije jednoznačan i, osim kod sasvim jednostavnih programa, nije ni malo lak. Vidimo na primer da smo u programu koji smo pisali označili tzv. *labelama* (o njima docnije) tri mesta koja smo nazvali „petlja“, „kraj“ i „poruka“ pri čemu ti nazivi otprilike asociraju na namenu oznaka. Ukoliko neko disasemblira ovaj program neće nikako moći da zna za ove naše oznake pa će mu praćenje čitave stvari biti daleko komplikovanije. Zato uvek i obavezno čuvajte na traci ili disketi izvorne verzije svih programa koje napišete: te verzije će vam biti dragocene ako jednom poželite da promenite neki svoj program ili da ga uklopite u nešto drugo.

Vrsta asemblera

Došlo je vreme da se pozabavimo vrstama asemblera koje ćete sresti kod kućnih računara. Najjednostavniji (i paradoksalno, najkorisniji, ne samo za početnike) asembleri su uklopljeni u bejzik: asemblerske instrukcije pišete kao obične bejzik naredbe, a zatim ispred i iza njih navodite

nešto (npr. otvorenu i zatvorenu uglastu zagradu) što će objasniti računaru da treba da ih prevodi a ne da ih izvršava. Zbog čega je to zgodno? Pre svega, asembler je u ROM-u zajedno sa bejzik interpretatorom pa vam je stalno pri ruci. Ne morate, osim toga, da učite da upotrebljavate neki novi editor: mašinske programe kucate i ispravljate poput bejzika. Da stvar bude posebno lepa, možete lako da kombinujete bejzik program koji će da učitava podatke i štampa rezultate sa asemblerskim potprogramima koji će se pobrinuti da spora izračunavanja postanu brza.

Druga mogućnost je asembler koji se sa svojim editorom upisuje u memoriju sa trake ili diska. Mana ove koncepcije je što kada vam program krahira i izbriše memoriju računara (a to će vam se, verujte nam na reč, vrlo često dešavati) ne samo dok učitaj morate da učitavate ne samo izvorni program već i čitav asembler koji je dug nekoliko kilobajta. Ovaj problem postaje posebno akutan ako sve treba da se učita sa trake! Za uzvrat imate editor koji je posebno prilagođen radu sa asemblerom i koji eventualno poseduje mogućnosti makro ekspanzija (možete da upotrebljavate svoje „mašinske naredbe", koje se pre asembliranja automatski zamenjuju nizovima pravih mašinskih naredbi koje ste ranije definisali). Ukoliko se opremite ovakvim asemblerom moraćete, naravno, da naučite da koristite jedan nov editor što možda i nije tako loše.

Poslednji i uslovno rečeno najprofesionalniji asembler programi uzimaju izvornu verziju sa diska, a zatim prevedeni kod smeštaju na isti medij. Šta se time dobilo? Pre svega treba da primetimo da se instrukcija LD A, (&1000) sastoji od petnaestak znakova koji se prevode u svega tri bajta objektnog koda. To znači da je izvorni zapis programa, pogotovu kada ga dopunimo komentarima, višestruko duži od konačnog programa koji pišemo. Ukoliko naš računar ima 48 Kb RAM-a od kojih 8 odu na sadržaj ekrana, sledeća četiri za program koji smo nazvali asemblerom i još par kilobajta za razne sistemske promenljive, ostaje nam tridesetak kilobajta. Ukoliko zaista želimo da napišemo toliko dugačak mašinski program, njegov izvorni oblik će biti dug stotinak kilobajta, što znači da nema nikakve šanse da stane u memoriju. Kapaciteti disketa su sa druge strane daleko veći pa nema mnogo problema da se naš izvorni program rastegne na stotinak kilobajta! Naravno, ovakvi problemi kao i asembleri koji ih rešavaju prilično su daleko od čitalaca ovog umetka pa ih zato nećemo podrobnije razmatrati.

Monitori

Dalji asemblera, za učenje mašinskog jezika će vam dobro doći program koji se naziva *monitor* ili *debager*. Ovaj program vam omogućava da pretražujete memoriju i menjate njen sadržaj i, što je posebno važno, izvršavate mašinske programe instrukciju po instrukciju posmatrajući promene u registrima i memoriji. Iskusni programeri koriste ovakve programe za traženje grešaka i tako štede svoje vreme dok će za početnike svakako biti interesantna mogućnost da u nekoj vrsti „usporenog filma" proučavaju delovanje naredbi koje upoznaju. Reći ćemo ipak da upotreba monitor programa obično nije sasvim jednostavna i da se njima treba da se upoznate tek kada bar pojmovno shvatite osnove mašinskog programiranja.

Pre nego što pređete na čitanje sledećih poglavlja koja će se baviti naredbama mikroprocesora Z80 treba da rešite par praktičnih problema koji će proisteći iz činjenice da programi koje ćemo sastavljati mogu da se izvršavaju na računaru, što znači da je poželjno da neke od njih i otkucate. Pre toga morate da proverite na koje memorijske adrese treba da smeštate mašinske programe kako se oni ne bi „kosili" sa asemblerom i izvornim kodom (ako se, na primer, asembler nalazi u memorijskim ćelijama čije su adrese &3000—&4000, direktiva ORG &35000 će skoro sigurno izazvati krah čitavog sistema jer će asembler asemblirajući program uništiti samoga sebe). Treba, osim toga, da proverite kako se mašinski programi pozivaju iz bejzika (za to obično služi naredba CALL ili funkcija USR) i kako se iz njih vraća u bejzik (obično mašinskom naredbom RET).

Arhitektura Z80

U uvodnom smo poglavlju objasnili pojmove *registar* i *fleg*, došlo je vreme da te pojmove upoznamo i na konkretnom primeru — arhitekturi mikroprocesora Z80.

Z80 je opremljen sa 14 opštih registara i osam registara specijalne namene, ovaj broj je, kada se radi o osmobitnom procesoru, prilično impresivan (6502, na primer, ima samo pet registara) što znači da vešt programer na Z80 može da izbegne prečesto pristupanje memoriji koje produžava i usporava program. Naravno, da bismo postali vešti programeri koji će postići to skraćenje i ubrzanje, treba dobro da upoznamo mnogobrojne registre koje nam je Zilog poklonio.

Registri specijalne namene

Upoznati svrhu registara opšte namene nije teško — u njih smeštamo podatke sa kojima operišemo. Jedan od tih registara (A) smo već upoznali — u njemu se obavlja većina aritmetičkih i logičkih operacija. Ostali registri opšte namene se obeležavaju slovima B C D E H i L. Svaki od njih može da sačuva po jedan osmobitni broj pri čemu se za neke instrukcije ovi registri kombinuju u šesnaestobitne BC, DE i HL. Zanimljivo je reći da je Zilog uspeo da ovim šesnaestobitnim registrima, čija su imena dobijena prostim ređanjem slova abecede, dodeli specijalna značenja. BC je skraćenica od Byte Counter ili brojač bajtova, nešto kao konjmina

Sabiranje osmobitnih brojeva

Postoje dve naredbe za sabiranje osmobitnih brojeva: ADD i ADC. Obe imaju po jedan argument koji naznačava broj koji treba dodati sadržaju akumulatora, pri čemu se rezultat opet smešta u akumulator. U čemu je onda razlika između ove dve naredbe? ADD B će dodati sadržaj registra B na registar A, dok će ADC B na sadržaj registra A dodati registar B **i stanje C flega**. Kako se dodaje stanje nekog flega? Ništa lakše: ako je fleg setovan, vrednost mu je jedan, a ako je resetovan onda mu je vrednost nula. Posmatrajmo, dakle, sledeća dva programa:

```
                          SCF
LD     B,&20              LD     B,&20
LD     A,&40              LD     A,&40
ADD    B                  ADC    B
```

Prve dve LD naredbe u oba programa treba da dovedu konstante &20 i &40 u registre B i A, a zatim sledi njihovo sabiranje. Pitanje je, naravno, šta će se po izvršavanju programa dobiti u akumulatoru. Što se prvog primera tiče, stvar je jednostavna: u akumulatoru će biti &20+&40+**CY** gde smo sa **CY** označili stanje indikatora prenosa (carry flega). Kakvo je to stanje? Obratimo pažnju na prvu naredbu drugog programa SCF; ova reč je skraćenica od *set carry flag* što bi moglo da se prevede i kao *stavi 1 u CY (carry)*. CY će, dakle, biti 1, pa će izvršavanjem drugog programa u akumulatoru biti &20+&40+1=&61!

Šta bi se desilo da smo izostavili naredbu SCF? Reklo bi se da je tada stanje indikatora prenosa nedefinisano, ali ono jednostavno ne može da bude takvo: fleg može da bude ili setovan ili resetovan, trećega nema! Zato će izvršavanjem prethodnog dela programa (ili radom bejzik interpretera ako je naš mašinski program upravo pozvan) biti formirana vrednost CY, pa će naredba ADC raditi korektno iako možda ne onako kako smo želeli. Zapamtite da, ako planirate korišćenje naredbi ADC i SBC (subtract, oduzimanje), **uvek** setujete indikator prenosa primenom SCF ili ga resetujete naredbom AND A (zašto ove naredbe resetuju indikator prenosa? Zato što tako piše u tablici. Zašto ne menja ništa drugo što bi bilo bitno? Malo strpljenja!)

Videli smo kako se izvršavaju naredbe ADD i ADC, ali nam je i dalje nejasno kakav je smisao kvarenja lepo dobijenog zbira dodavanjem indikatora prenosa. Zamislimo da treba da saberemo dva šesnaestobitna broja od kojih se jedan nalazi u memorijskim ćelijama &2000 i &2001, a drugi u &2002 i &2003. Zbir treba da smestimo u &200A i &200B. Obzirom da smo rekli da Z80 ima naredbe za šesnaestobitno sabiranje, problem se lako rešava. Te naredbe, međutim, još nismo upoznali pa nećemo ni da ih koristimo, one nam, uostalom, ne bi mnogo pomogle da treba da sabiramo trideset dvobitne ili veće brojeve. Kako da rešimo problem? Pokušaćemo da najpre saberemo sadržaje memorijskih ćelija &2000 i &2002 (niže bajtove oba broja) i rezultat smestimo u &200A, a da zatim saberemo sadržaje &2001 i &2003 i rezultat smestimo u &200B. Evo programa koji to radi:

```
LD     A,(&2002)
LD     B,A          pošto naredbe LD B,(&2002) ne postoji
LD     A,(&2000)
ADD    B
LD     (&200A),A    sabrali smo manje signifikantne delove brojeva.
LD     A,(&2003)
LD     B,A
LD     A,(&2001)
ADD    B
LD     (&200B),A    sabrali smo više signifikantne delove brojeva
```

Isprobajmo ovaj program stavljajući u &2000 i &2001 broj &2030 (POKE &2000, &30: POKE &2001, &20) a u &2002 i &2003 broj &4050. Posle startovanja programa dobićemo da se u ćeliji &200A nalazi broj &80 (ako iskoristite PRINT PEEK (&200A) dobićete 128, što ne treba da vas iznenadi: 128=&80) a u ćeliji &200B broj &60, što znači da je zbir &6080: program, dakle, radi ispravno. No šta ako u ćelije &2000, &2001, &2002 i &2003 upišemo respektivno brojeve &20, &3A, &F0 i &10? Po startovanju programa će se dobiti zbir &4A10 dok je &3A20+&10F0 = &4B10. Program je, dakle, pogrešan!

Kako je nastupila greška? Z80 je najpre sabrao brojeve &20 i &F0 i dobio, ili bar trebao da dobije, &110. Broj &110 je, međutim, veći od &FF, pa se pojavio bit koji nije mogao da stane u akumulator. Onako kako smo ga napisali, program je zanemario taj bit pa je sabrao brojeve &3A i &10 i dobio &4A; da je dodao prenos, dobio bi &4B što bi bio tačan rezultat.

Problem ćemo rešiti tako što ćemo drugu ADD instrukciju zameniti sa ADC B i sve će biti u redu. Kako je to moguće? Osim formiranja rezultata u akumulatoru, naredbe ADD i ADC menjaju sadržaje nekih flegova od kojih je jedan i CY. Ako se pojavi prenos, to jest ako je rezultat sabiranja veći od &FF, indikator prenosa biva setovan, dok je u suprotnom resetovan. Na taj način će ADC instrukcija uzeti u obzir prenos od prethodnog sabiranja i formirati korektan rezultat. Dataje još da objasnimo razlog zbog koga nismo obe ADD instrukcije zamenili sa ADC. U tom slučaju bismo morali da resetujemo indikator prenosa kako bismo osigurali korektno obavljanje prvog sabiranja: ako bi CY bio setovan, dobili bismo za jedan veći rezultat od korektnog. Mogli smo, naravno, da napišemo jedno AND A (naredba kojom kako rekosmo resetujemo CY), ali bi to potrošilo bajt programa više!

Bilo je kao i obično više priča nego programa. Pogledajmo ipak kako sve to radi ako je korisnik tražio da mu se prikaže sadržaj ćelije čije je adresa #3333 u registru HL će se po povratku iz potprograma TAST nalaziti broj &3333. LD C, (HL) je jsano specijalni slučaj instrukcije LD reg (HL) gde reg predstavlja bilo koji registar. Primenom ove ćelije se *u registar C dovodi sadržaj memorijske ćelije čija je adresa u HL* u našem slučaju ćelija &3333. Taj broj će primenom potprograma ISPIS biti napisan na ekranu.

Za indirektno adresiranje kod Z80 se uglavnom koristi registar HL koji smo jednom nazvali *šesnaestobitnim akumulatorom* posedujemo čak i instrukciju LD (HL) nn kojom možemo da upisujemo konstante direktno u memoriju. Primetimo da instrukcija Ld (nnmm) nn ne postoji što znači da ne *možemo* da napišemo nešto poput LD (&2000) 0 moraćemo da upotrebimo LD A 0 LD (&2000) A ili LD HL (#2000) LD (HL) 0 zavisno od toga da li nam je draže da obnšemo pretkodni sadržaj akumulatora ili prethodni sadržaj registra HL. Indirektno smeštanje akumulatora u memoriju ili farmiranje njegovog sadržaja možemo da postignemo i primenom instrukcija LD (BC),A odnosno LD (DE),A i njihovih ,ogledalskih simetrija LD A (BC) odnosno LD A (DE). Suvišno je reći da *ne postoje* instrukcije LD B (DE) LD H (BC) i slične. Sve instrukcije indirektnog adresiranja ili kako bi ga puritanci nazvali indirektnog *registarskog* adresiranja kod Z80 zauzimaju samo po jedan bajt.

Indeksirano adresiranje primenom registara IX i IY je daleko sporije i troši daleko više memorije pa se retko koristi. Posedujemo instrukcije LD (IX+dd) reg LD reg (IX+dd) i LD (IX+dd) nn kao i odgovarajuće instrukcije koje umesto IX koriste IY. Izvršavaju se tako što se sadržaj IX (ili IY) sabere sa brojem *dd* koji se nalazi u okviru same instrukcije i tako se dobija adresa ćelije kojoj treba pristupiti. Najveća vrednost za *dd* je 255.

U osmobitne load instrukcije spadaju i one koje prenose sadržaj akumulatora u specijalne registre I i R i koje omogućavaju njihovo čitanje tj. prenos vrednosti u akumulator. Za sada vam ove instrukcije neće biti naročito značajne.

Šesnaestobitne load instrukcije

Kao što smo se *reg* obeležili bilo koji osmobitni registar opšte namene sa *rp* ćemo obeležiti ulio koji od registarskih parova BC DE HL ili specijalni registar SP. Bilo koji od ovih registara možemo da popunimo proizvoljnom šesnaestobilnom konstantom koristeći instrukcije tipa LD rp nnmm i trošeći tri bajta programske memorije. U prvi se smešta kod instrukcije a u preostala dva šesnaestobitna konstanta *nnmm* upisana kao i obično, tako da se na nižoj adresi nalazi niži bajt. Pogled na tačelu iz našeg prošlog umetka će pokazati da se LD HL #2DCC kodira kao &21 #CC &2D. Za punjenja registara IX i IY konstantama smo snabdeveni naredbama LD IX nnmm i LD IY nnmm.

Potrebne su nam naravno i instrukcije za prenošenje sadržaja memorija u šesnaestobitne registre. Opšti oblik takve naredbe je LD rp (nnmm), njenim izvršavanjem se u registar *rp* dovode sadržaji memorijskih ćelija čije su adrese *nnmm* i *nnmm+1*. Naredba LD BC (&2DAA) će na primer dovesti u C sadržaj ćelije #2DAA a u B sadržaj ćelije &2DAB. Instrukcije ovoga tipa zapremaju čitava četiri bajta memorije dva za njihov kod i dva za adresu ćelija. Konstruktori Z80 su obezbedili i specijalnu naredbu LD HL (nnmm) koja zauzima samo tri bajta jsano je da je LD HL (#2000) specijalni slučaj naredće LD rp (nnmm) i da može da se kodira kao #ED &6B &00 &20 ali se ista naredba može kodirati i kao #2A &00 #20 zauzimajući bajt manje. Vaš asembler će ako iše vredi uvek automatski izabrati ovaj drugi oblik kodiranja.

Pošto ste polako počeli da razumevate strukturu LD instrukcija neće vam biti teško da pogodite da je obezbeđena instrukcija LD (nnmm) rp i kao njen specijalni slučaj LD (nnmm) HL, ove se naredće koriste za smeštanje sadržaja šesnaestobitnih registara u memoriju. Naredba LD IX (nnmm) i LD (nnmm) IX (iste takve postoje i za IY) se koriste za rad sa sadržajima indeksnih registara.

Na prvi je pogled dosta čudno što nas Zilog *nije* opremio instrukcijama tipa LD rp rp kojima bismo mogli da razmenjujemo sadržaje šesnaestobitnih registara (nemamo na primer naredbu LD HL DE). Pokazuje se međutim da ove instrukcije nisu naročito potrebne jer mogu da se simuliraju uz utrošak svega dva bajta programa. LD HL DE može da se zameni sa LD H D LD L E! Registar SP je izuzetak od ovoga pravila kako ne možemo da pristupamo njegovim osmobitnim komponentama snabdeveni smo naredbama LD SP HL LD SP IX i LD SP IY *ali ne* i naredbom LD HL SP.

U šesnaestobitne load instrukcije spadaju strogo uzevši i naredbe PUSH i POP kojima se operiše sa stekom ali ćemo se njima pozabaviti u posebnom poglavlju.

Aritmetičke i logičke operacije

U drugom poglavlju smo upoznali operacije sa binarnim brojevima koje Z80 zna da izvrši sabiranje i oduzimanje. Poput prenošenja podataka sabiranje i oduzimanje može da bude osmobitno ili šesnaestobitno dok se tek primenom malih mašinskih programa može operisati sa većim brojevima.

promenljiva u FOR NEXT petlji bejzika. DE je skraćenica za Destination (odredište): ovaj registar pri prenošenju blokova memorije sadrži adresu početka memorijskog bloka koji će da primi podatke. Ime registra HL (HighLow) treba jednostavno da nas podseti da će se posle upisivanje broja &20 u registar H i broja &AA u registar L u registarskom paru HL nalaziti broj &20AA a ne broj &AA20. Uskoro ćemo videti da se najveći broj osmobitnih operacija obavlja nad registrom A a najveći broj šesnaestobitnih nad registrom HL koji predstavlja neku vrstu šesnaestobitnog akumulatora.

Alternativni registri

Upoznali smo registre A B C D E H I L svega njih 7. Na početku ovog poglavlje smo međutim rekli da Z80 ima 14 registara opšte namene. Preostalih 7 registara nazivamo *alternativnim* i označavamo sa (prim) iza imena. Tako postoje registri A B C D E H I L kao i registarski parovi BC DE i HL (obratite pažnju da se registar DE sastoji od registra D i E a ne registara D i E pravilnije bi ga možda bilo zvati registar D E). Mikroprocesor u jednom trenutku može da opšti samo sa jednim setom registara ali postoji naredba EXX koja će razmeniti sadržaje B C D E H i L sa sadržajima B C D E H i L i tako učiniti alternativna registre pristupačnima a osnovne nepristupačnima. Što se registara A i A tiče njih razmenjujemo naredbom EX AF AF pri čemu se razmenjuje i registar F koga smo ubrojili u registra specijalna namene i o koma ćemo govoriti malo docnije.

Za šta mogu da se iskoriste alternativni registri? Zamislimo da smo sastavili potprogram koji će se pozivati izuzetno često (npr. 50 puta u sekundi) i koji obavlja neku operaciju nezavisnu od glavnog programa kao što je crtanje slike na ekranu našeg monitora (videćemo da ove uslove zadovoljavaju programi za opsluživanje interapta). Tom programu je za rad neophodno nekoliko registara a istovremeno na smemo da dopustimo da se promena vrednosti koje je u ta registre postavio glavni program pre poziva potprograma. Mogli bismo naravno da sačuvamo sadržaje registara na steku ili ih prenesemo u fiksne memorijske ćelije. Daleko je međutim kraće i brže da na početak potprograma i na njegov kraj stavimo instrukciju EXX koja će sačuvati odnosno restaurirati sadržaje opštih registara. Postojanje alternativnih registara nam praktično omogućava paralelno izvršavanje dva nezavisna programa.

Registri opšte namene

Registri specijalne namene se označavaju sa PC SP IX, IY F I i R. Svi registri čija se imena sastoje od po dva slova su kao i do sada šesnaestobitni pri čemu u setu instrukcija nije ostavljena mogućnost pristupanja njihovim osmobitnim komponentama (ne možemo na primer da pristupimo registru X koji je deo registra IX iz log razloga nije opasno postojanja imena I IX i IY jer se zajedničko i u njima na odnosi na isti registar). Registri R i F nisu kao što bi se moglo očekivati osmobitni registru R nedostaje sedmi najviši bit dok registru F nedostaju bitovi 3 i 5. Razloge za ova anomalija ćemo shvatiti u vrlo skoroj budućnosti pošto obradimo normalne registre PC IX IY i SP.

Programski brojač

U setu instrukcija naćete naći ni jednu jedinu koja operiše sa sadržajem registre PC a ipak se taj sadržaj neprekidno menja. PC je skraćenica od *Program Counter* ili u bukvalnom prevodu programski brojač. U ovom se registru nalazi adresa instrukcije koju mikroprocesor traba da obredi tj. adresa bajta koji će pri normalnom izvršavanju programa biti sledeći pročitan. Kako se neke instrukcije sastoje od više bajtova registar PC će u toku njihovog izvršavanje nekoliko puta menjati vrednost. Pretpostavimo da je na redu izvršavanje instrukcija LD A (&4000) koja je kodirana kao &3A &00 #40 i kao što vidimo zauzima tri bajta memorije. &3A je oznaka instrukcija LD A (nnmm) a #00 i &40 predstavljaju adresu &4000 koja se kao i uvek prikazuje sa nižim bajtom napred. Pre izvršavanje instrukcije u PC će biti upisane adresa memorijska ćelije koje sadrži broj #3A — početak instrukcije koja se obrađuje.

Mikroprocesor najpre uzima kod instrukcije (&3A) i smešta ga u registar naredbi koji je za nas nevidljiv. Istovremeno se registar PC povećava za 1 tako da pokazuje na ćeji #00. Mikroprocesor prepoznaje instrukciju LD A (nnmm) i primećuje da mu je potrebna šesnaestobitna adresa pa zahteva učitavanja njenog nižeg bajta na koji pokazuje PC. U registar adrese (ponovo nevidljiv za nas) dolazi broj &00 a PC se povećava za 1 tako da pokazuje na broj &40. Sledi još jedno pristupanja memorija i prenošenje broja &40 u registar adrese pri čemu se PC ponovo povećava i pokazuje na sledeću instrukciju koju mikroprocesor treba da izvrši. Pre nego što dođe do nje mikroprocesor će jsano morati još jednom da pristupa memoriji da bi pročitao sadržaj ćelija &4000 i smestio ga u akumulator.

Iako to u setu instrukcija nije izričito spomenuto možemo lako da upisujemo brojeve u registar PC instrukcija JP #3A00 (JUMP) bi slobodno mogla da se zove i LD PC #3A00. Posle izvršavanja ove instrukcija mikroprocesor će nastaviti normalan red ali neće izvršavati sledeću naredbu već onu čiji je broj nasilno upisan u PC — u našem slučaju naredbu čiji je početak upisan u ćeliju &3A00. Omogućeno nam je i da dodajemo konstante sadržaju registra PC takva dodavanja smo već upoznali pod imenom *relativni skokovi*.

Pokazivač steka

Registar SP *(Stack pointer* ili pokazivač steka) ima nekih sličnosti sa registrom PC njega vrlo retko direktno menjamo (obično se takve operacije obavlja negde na logičnom početku ROM-a) ali su njegove implicitne promene vrlo česte. Ovaj registar pokazuje na neku memorijsku ćeliju ispod koje ima (uslovno rečeno) dovoljno prostora u RAM-u. Kada nam zatreba da privremeno sačuvamo sadržaj nekog **registarskog para** (Z80 je koncipiran tako da se na stak smeštaju samo parovi registara ne pitajte nas zašto) upotrebljavamo instrukciju poput PUSH HL. Tada će SP biti umanjen za 1 sadržaj registra H prekopiran u memorijsku ćeliju na koju pokazuje SP zatim će SP biti ponovo umanjen za 1 i na kraju će sadržaj registra L biti prekopiran u memorijsku ćeliju na koju ukazuje SP. Potpuno suprotnu proceduru će izvršiti naredba POP HL koja će kao krajnji rezultat preneti u registar HL dva bajta sa vrha steka i povećati SP za dva.

Vidimo da je stek na neki način suprotan programu registar PC se povećava kada obradimo instrukciju što znači da program „raste" od nižih ka višim adresama. Suprotno tome sadržaj SP se smanjuje kada stavimo broj na stek što znači da ovaj „raste" ka nižim adresama. Zahvaljujući tome program možemo da upisujemo od „dna" memorije a za stek da rezervišemo njen „vrh". Kada SP pokaže na zadnju adresu programa stek i program su se „sudarili" pa nema više memorije za normalan rad računar bi tada trebao da prijavi grešku ali je mnogo verovatnije da će nastupiti krah.

Indeksni registri

IX i IY su indeksni registri neka instrukcija u adresnom delu mogu da imaju oznaku (IX+d) ili (IY+d) što znači da se adresa memorijske ćelije kojoj treba pristupiti dobija sabiranjem sadržaja registra IX (IY) i fiksnog ofseta *(displacement)* d koji se nalazi u okviru same instrukcije. Ovakvo indeksiranje je prilično beskorisno i njime se načemo opširnije baviti registre IX i IY možete da koristite i kao registre opšte namene (za to su sasvim dobri) ali prethodno provarite da takvom rešenju nisu pribegli konstruktori ROM-a vašeg računara nije malo kompjutera kod kojih IX i IY imaju neku specijalnu funkciju (npr pokazivači aritmetičkog steka) tako da ih mašinski programi ne smeju menjati ako planiraju regularan povratak u bejzik. Kod nekih računara (npr „galaksija") indeksnim registrima možete da pristupate samo ako **dobro** znate šta radite.

Osvežavanje memorija

Registar I se koristi za pamćenje višeg bajta takozvanog *interapt vektora* i ima ulogu samo u interapt modu 2.

Registar R je zadužen za automatsko osvežavanje *(refresh)* dinamičkih memorije — karakteristiku koje Z80 stavlja iznad većine procesora koji za ovu neprekidno potrebnu operaciju zahtevaju dodatni hardver. Ovaj registar se automatski povećava posle izvršavanja svake instrukcije i nekoristan je za programera koji nema hardverskih ambicija.

Registar uslova

Došli smo, najzad do registra F kome ćemo posvetiti dužnu pažnju. Već smo rekli da mikroprocesor poseduje određen broj flegova koji se koriste za kontrolu toka programa i neke aritmetička i logičke operacije. Svi ovi flegovi su fiktivno svrstani u registar F kako bi njihovo stanje moglo da se sačuva na steku ili alternativnom registru F (registar F je uparen sa registrom A pa tako postoje naredba PUSH AF i POP AF kao i naredba EX AF AF koju smo već pominjali).

Bit nula F registra zove se indikator prenosa (Carry) i naravno obeležava slovom C koje ne treba mešati sa imenom jednog od registara. Ovaj bit se setuje kada se pri binarnom sabiranju pojavi prenos iz najstarijeg razreda (sabiramo na primer &33 i &FA pokušajte) ili kada oduzimamo veći broj od manjeg. Postojanje ovoga flega kao što ćemo videti omogućava jednostavno sabiranje i oduzimanje brojeva koji su smešteni u nekoliko memorijskih ćelija. C fleg isto tako, predstavlja deveti bit akumulatora pri šiftovanjima pa omogućava prenošenje bitova iz jednog broja u drugi.

Bit jedan registra F se obeležava sa N i interno se koristi pri izvršavanju naredba DAA. Za programera je nekoristan.

Bit dva — indikator parnosti/prekoračenje — se označava sa P/V (od *Parrity/Overflow*) i ima vrlo složenu funkciju. Kod logičkih operacija i IN instrukcija ovaj fleg je setovan ako je broj binarnih jedinica u rezultatu paran (Parrity). Nevezano sa tim P/V će biti setovan ako je rezultat prethodnog sabiranja u akumulatoru nakorekian usred prekoračenja pogledajte drugo poglavlje i sabiranje označenih brojeva — P/V fleg u ovom slučaju praktično predstavlja deveti bit akumulatora) a resetovan ako je rezultat korektan (Overflow). P/V fleg se menja i kod instrukcija koje pomeraju blokove memorije kao i pri promeni sadržaja i registra ali vas te finese za sledeće duže vreme neće interesovati.

Bit tri registra F se ne koristi.

Bit četiri — indikator poluprenosa — poznat je pod imenom *Half Carry* i označen sa H ali je za programera nekoristan. Mikroprocesor se služi ovim flegom pri izvršavanju već inkriminisane naredbe DAA.

Bit pet registra F se ne koristi.

Bit šest je konačno značajan obeležavamo ga sa Z *(Zero)* i zovemo indikator nule. On se automatski setuje kada je rezultat neke operacije nula ili kada je test dao rezultat True (tačno) i kao takav je neobično koristan za realizaciju strukture koje odgovaraju bejzik naredbi IF A=B THEN.

Znajući da bit sedam svakog bajta nosi znak broja nećete se iznenaditi kada pročitate da se bit 7 registra F zove indikator znaka *(Sign flag)* i obeležava sa S. On je setovan kada je rezultat neke operacije (npr oduzimanja) negativan tj ako mu je najviši bit setovan. Ovaj fleg se koristi za realizaciju mašinskih struktura koje odgovaraju bejzik naredbi AF A < B THEN.

LD grupa

Prva grupa instrukcija mikroprocesora Z80 kojom ćemo se baviti su LOAD instrukcije koje su zadužene za prenošenje podataka između memorije i registara. Neke od tih instrukcija smo već upoznali u dosadašnjim primerima smo često koristili nešto poput LOAD A &20 što je trebalo da znači *dovedi broj &20 u registar A*. Da sada budemo malo egzaktniji u mnemonici Z80 naredba LOAD se piše kao *LD A B* i znači *unesi (LOAD=napuni) u A sadržaj registra B*. Pročitajte ovu rečenicu još jednom. *Naredbom LD se sadržaj **drugog** argumenta prenosi u **prvi** argument.*

A i B mogu jasno da budu raznorazne stvari osmobitni i šesnaestobitni registri memorijske ćelije adresirane direktno indirektno ili i indeksno i naravno obične konstante. Bitno je međutim da zapamtite da A i B *mogu* da budu svašta ali *ne mogu* da budu bilo šta neke instrukcije postoje a druga ne! Tako na primer možete da napišete LD A (&2000) ali *ne možete* da napišete LD B (&2000). Jasno je da su ove dva instrukcije prilično slične i da bi svaka od njih imala smisla zašto u registar B ne bismo smeli da dovedemo sadržaj memorijske ćelije &2000? Samo zato što su konstruktori mikroprocesora Z80 bili upućeni na stabilizaciono ponašanje pa u set instrukcija nisu uveli sve moguća varijante već samo one koje su smatrali za najkorisnije. Zbog ove nesimetričnosti seta instrukcije opširno ćemo se baviti svim regularnim varijantama na koje ćete vas docnije podsećati i tabela iz našeg prethodnog umetka.

Osmobitne load instrukcije

Osmobitne load instrukcije se kao što im ime i govori bave prenošenjem bajtova (osmobitnih brojeva) između registara i memorije. U tabeli vidimo da je prva moguća naredba ovoga tipa LD reg reg gde je reg oznaka bilo kog od registara A B, C, D E H i L. To znači da slobodno možemo da pišemo LD A B LD E H LD H E i sve slične kombinacije koje nam zatrebaju. U principu bismo mogli da koristimo instrukcije tipa LD A,A ili LD B B koje jasno nemaju nikakvog smisla ali su dozvoljene. Sva instrukcije tipa LD reg reg zauzimaju po jedan bajt memorije i ne utiču na stanje flegova kao uostalom i sve ostale LD naredbe koje ćemo koristiti.

Instrukcije LD reg nn predstavljaju prototip neposrednog adresiranja koje smo već upoznali u bilo koji od gore pomenutih 7 registara koje smo označili sa reg primenom ovakvih naredbi se dovodi osmobitne konstanta *nn*. Naredba LD A &10 će na primer dovesti u registar A (akumulator) broj &10 tj 16. Instrukcije ovoga tipa zauzimaju po dva bajta u jedan se smešta kod a u drugi konstanta *nn*.

Primer apsolutnog adresiranja je LD A, (nnmm). Radi se o instrukciji koja zauzima tri bajta memorije, u prvom se nalazi njen kod &3A a u preostale dva šesnaestobitni broj *nnmm* koji predstavlja adresu neke memorijske ćelije. Dejstvo naredbe je lako objasniti u akumulator se dovodi sadržaj memorijska ćelije čija je adresa *nnmm*. Protivteža ovoj naredbi je LD (nnmm) A čijom primenom smeštamo sadržaj akumulatora u memorijsku ćeliju čiji je broj *nnmm*. Primetimo da se ove instrukcije *ne navode* kao LD reg (nnmm) ili LD (nnmm) reg što znači da jedini registar sa kojim operišemo sme da buda akumulator, vaš asembler će prijaviti grešku ako pokušate da prevedete naredbu kao što je LD B (&2000).

Posle apsolutnog upoznaćemo indirektno adresiranje. Konstruktori Z80 su nam obezbedili instrukcije LD reg (HL) LD (HL) reg LD (HL) nn i specijalne naredba LD A (BC) LD A (DE) kao i njihova ogledalska oblike LD (BC),A i LD (DE) A. Iako smo se jednom bavili indirektnim adresiranjem uzećemo slobodu da ga opišemo još jednom jer se radi o važnoj stvari koju svaki početnik ne mora odmah da razume. Nećemo se naravno baviti svim instrukcijama, već samo jednom LD reg (HL).

Pretpostavimo da treba da sastavimo program koji će omogućavati korisniku da pregleda sadržaj memorije. Korisnik našeg programa će na primer otkucati broj neke memorijske ćelije (između 0 i &FFFF) a naš će program na ekranu ispisati njen sadržaj. Za sada jasno ne umemo da napišemo potprogram za učitavanje adrese ćelije sa tastature i ispisivanje njenog sadržaja na ekranu ali možemo da pokušamo da napišemo ono što se nalazi između poziva ta dva potprograma.

CALL TAST	Poziv potprograma koji sa tastature učitava šesnaestobitni broj i smešta ga u registar HL
LD C (HL)	Prenošenje sadržaja tražene ćelije u C
CALL ISPIS	Poziv potprograma koji ispisuje sadržaj C na ekran

Mali oglasi

Ako ne možete da podnesete da drugi nemaju ono što vi imate objavite svoj mali oglas u „Računarima"

Ako ne možete da podnesete da drugi imaju ono što vi nemate, javite se na neki od malih oglasa u „Računarima"

Ako ne volite da se dopisujate sa „Računarima", svoj mali oglas možete nam izdiktirati preko telefona 011/650-161 svakog radnog dana od 10—14 sati Mi ćemo vam onda naknadno poslati ispunjenu uplatnicu

Prva stvar koju treba da uradite je da se odlučite da li želite običan ili uokviren mali oglas

*CENA OBIČNOG MALOG OGLASA do dvadeset reči je 900 dinara Svaka naredna reč košta još 60 dinara, s tim što **oglas ne sme da ima više od 50 reči.** Adresa oglašivača se ne računa u cenu*

CENA UOKVIRENOG MALOG OGLASA je 900 dinara po visinskom centimetru, s tim što se mogu zakupiti najmanje 32 slovna znaka Ako se ne iskoristi čitav prostor u jednom radu, računa se broj redova a ne broj znakova Za uokvirene oglase preko 5 cm cena je 1400 dinara po centimetru

Poželjno je da vaš mali oglas počinje sa Prodajem, Kupujem, Držim časove, Menjam ... ili načim sličnim što ukratko ukazuje na sadržaj oglasa

Da ne bi bilo zabune, obavezno naznačite da li želite običan ili uokvireni mali oglas, i zajedno sa tekstom vašeg malog oglasa pošaljite i priznanicu o uplati na adresu redakcije GALAKSIJA, BULEVAR VOJVODE MIŠIĆA 17 BEOGRAD, sa naznakom „za male oglase u RAČUNARIMA"

Svi mali oglasi koji stignu u našu redakciju do 22 maja do 12 sati biće objavljeni u „Računarima" broj 16, koji izlazi iz štampe polovinom juna

SPEKTRUM

□ Igra Winter Games Kung Fu Beach Head Willy 4 Movie A.O Yesod Pyjamarama 4 Blade Runner Summer Games Elite Gunfright Hyperblaster prodajam za 600 dinara I drugi kompleti Saša Radojković 3 oktobar 165 19210 Bor tel 030/38-182

□ Spektrumovci za vašu dugu prodajam paket programa po ceni od 1000 din bez kasete i to B C Quest For Tires Sir Fred Winter Games Arc Of Yesod Zorro Gunfright Beach Head 3 Jet Set Willy 4 Pyjamarama 4 Mojsin Zoran Al Spomenice 5/38 19210 Bor tel 030/25-882

□ MOON CLUB — obratite se radi kupovine svih vrsta programa po vrlo povoljnum cenama Usluga brza i tačna a kvalitet zagarantovan Tražite katalog većine programa. Ne navodimo igre jer novi hitovi stalno pristižu Ne zaboravite naša adresa je Strika Jovan Grčića Milanka 4/a stan 135 11000 Beograd tel 011/444-5093

□ COMPUTER GAME SHOP vam predstavlja komplet ne baš super najnovijih ali najboljih i relativno najnovijih programa svih vremena za vaš spektrum Komplet+kaseta +poštarina samo 850 din Isporuku od 24 časa, kao i kvalitet snimka vam garantujemo 100%!! West Bank, Movie Barry Mc Gulgan Beach Head 3 Blade Runner Splitfire 40 Battle of Planets, Grumphy Supersleuth Tel 011/4887 575

□ Spektrumovci Biblioteka Softvera svojim članovima omogućava pristup u biblioteku od 500 programa sa spektrum mogućnost posudbe većine uputstava za programe pretplelu na najnovije programe koje dostavljamo u roku od 15 dana od pojave na tržištu pretplatu na kiupaku kompjuterski časopis New Bit, popuste pri nabavci hardvera i softvera kod najeminentnijih proizvođača ... i još mnogo drugog Tražite opširne besplatne prospekte Branko Čuić pp 57 47300 Ogulin tel 047/72 289 ili Milanko Savić M Tita 63/1 31000 T Užice tel 031/24-948

□ Prodajam ZX spektrum 48 K (izvrsno očuvan) sa pratećim priborom i uputstvima i nekoliko raznovrsnih programa Tel 034/212-009 Todorov Ratko Đorđa Andrejevića Kune 24 34000 Kragujevac

□ Loto i Sportska prognoza. Tri programa za spektrum ukupno 1500 din Iskoristite računar uvećajte svoje šanse za dobitak. Žarko Vukosavljević p fah 65 11070 N Beograd tel 011/197-700

□ Spektrumovci! Mega-prilika! Movie (pravi detektivski film) Three Weeks in Paradise (Pyjamarama 4) Barry McGuigan Boxing Gladiator (borba gladijatora), Tomahawk (helikopter) Forbidden Planet (čudo velikih majstora) Spellbound (budući klasik) Turbo Espirit (ludovanje po Njujorku) Arc Of Yesod (Nodes 2) Mugsy's Revenge (fantastično) za samo 800 din + kaseta. Nazovite, pišite jer mi radimo samo za vas Marko Premzl Rade Velna 5 43260 Križevci tel 043/841-682

□ SPECTRUM RAINBOW SOFTWARE vam nudi Needle Satancopy 4 Satancopy 3 Turbotape 1 Turbotape 2 Supercopy 1 Supercopy 2 Mastercopy+40 drugih copy programa u jednom kompletu za samo 1000 din Kompleti od 25 programa 800 din Posedujemo i sve najbolje programe koji se trenutno nalaze u Jugoslaviji Tražite besplatni katalog sa 2000 programa Snimanje direktno iz računara po najnižoj ceni! Uverite se! Mihajlovski Kirčo Moša Pijade 128 91300 Kumanovo tel 0901/23-800

□ Zodiac Strip — po oceni stručne štampe najbolji erotski program za spektrum Uzbudljiva igra sa deset devojaka. Cena 1000 din Žarko Vukosavljević p fah 65 11070 N Beograd tel 011/197-700

□ Spektrumovci GRUNF SOFT — prodajem najnovije i najbolje komplete za spektrum Komplet 30 Nomad Gladiator Gunfright ... Komplet 31 Saboteur Commando Eadel Pomo ... 12 programa+kazeta+poštarina 1200 din a sa vašom kazetom 600 din Za sve informacije obratite se na tel 041/271-025 ili 430-185

prijem malih oglasa zaključujemo 22. maja u 12 časove

□ Prodajam ZX spektrum 48 K, tastaturu Ines ZX Interfejs 1 ZX mikrodrajv štampač šinva CP-80 Interfejs kempston-S Tel 063/855-390 16-22h

□ Najnoviji programi za ZX spektrum u kompletu i pojedinačno Movie West Bank, Winter Games 1 i 2 Sky Fox Pyjamarama 4 Mikie Gunfright Wham The Music Box Tražite besplaten katalog Gulabovski Goran Lihnida 1/1 — 9 91060 Skoplje tel 091/314-186 ili 317-006

□ COMPUTER STUDIO nudi
Komplet 51 Rambo Impossible Mission Sex Mission
Komplet 52 Elite Mikie Commando
Uz to mnoge programe, literatura hardverski dodaci saveti pomoć
Četiri godine iskustva, nepovećavanje cena od 1983 preko 10 000 zadovoljnih mušterija iz zemlje i inostranstva govore nama u prilog
COMPUTER STUDIO Starec Bojan Kosančićev Venec 1A 11000 Beograd tel 011/625-833

□ Rečnik englesko-srpskohrvatski (oko 1400 najčešće upotrebljavanih reči)+kaseta+poštarina (1300 din) Tel 011/497-682 od 17—19h D Marjanović B Jerković 123 11000 Beograd

□ KRAPINKO SOFTWARE — najnoviji hitovi po najnižim cenama za ZX spektrum Programi u kompletima ili pojedinačno. Rok isporuke 48 sati Katalog besplatan Kiralj Denis Marka Oreškovića 1/7 55000 Slavonski Brod tel 055/238-666

□ Spektrumovci! Dosad neviđeni hitovi u kompletu 25 Tomahawk Beach Head 3 Type Rope Forbidden Planet Battle Of Planets Sky Fox Barry McGuigan Box, Code Name Mat 2 West Bank Movie Blade Runner Jetset Willy 4 Mega Fruit ... Cena 700 din.+kaseta Perić Nenad Braće Miladinov 12, 37000 Kruševac tel 037/33-510

COMET SOFTWARE
Programi za ZX SPECTRUM
video igre -
namenski programi
- uputstva za programe

BETA BASIC 3 0
sa kasetom
originalnim uputstvom
i poštarinom
cena 1000 din

garancija za sve usluge

MILOVANOVIĆ LJUBIŠA
Petra Lekovića 57 11030 Beograd
tel 011/558887 posle 17 h

□ Prodajem pojedinačno snimljene programe za spektrum Cena jednog programa je 100 din Na svakih pet naručenih programa jedan poklanjam! Želite li imati programe snimljene direktno iz kompjutera pišite za besplatan katalog! Janković Šifard Petefi Šandora 84 25222 Telečka

□ Spektrumovci Elite Rambo Hacker Commando XCL Nomad Saboteur Impossible Mission Cena jednog kompleta od 10 igara 600 din. Zoran Tonić, Fasila Španca 104/A 23261 Lukićevo

□ Spektrumovci veliki izbor programa. Cene 40 60 i izuzetno 60 dinara. Besplatna uputstva za igre Pokloni Razmene Tražite veliki besplatan katalog za igre D-2 SOFT 11420 Smed Palanka. Pionirska 15 tel 026/34-061

□ SPEKTRUM YU — SOFT ponovo sa vama Jedini koji (još uvek) snima direktno iz spektruma zato je snimak ispravan i posle nekoliko godina. Najnoviji program popust za pretplatnike besplatan spisak. JEREMIĆ NEBOJŠA, RISANSKA 10 11000 BEOGRAD tel 011/643-061

□ Super povoljno! Super programi po super niskim cenama. Super nagrade i super kompleti Tražite besplatan

□ ROLEX SOFT vam nudi najnovije programe po pristupačnim cenama u kompletima ili pojedinačno Sigurno imamo WS Kung Kong Movie Amazon Woman i sve što drugi imaju Najnoviji programi staju 200 a stariji 100 dinara. Spisak je besplatan a katalog košta 100 din Rosić Nebojša Čelebička 16/4 11000 Beograd tel 011/591-631

□ ADICTIVE SOFTWARE vam nudi najnovije hitove u kompletu ili pojedinačno po najpovoljnijim cenama za ZX spektrum The Way Of Tiger Boughdek F.A. Cup Football Amazon Women Green Bereth Enkrevobell Fireman In Action Yadbe Dabka Ooo Ping-Pong Visiters Endurances, Beach Head 3 Summer Games 2 Cena kompleta 600 din Ćirić Aleksandar Crnotravska 13 11000 Beograd tel 011/661-260

Svakog meseca najnoviji ZX spektrum programi u polusatnim kompletima (po 6 programa) za samo 200 din. Tražite katalog sa 1000 programa. Marko Merinović. Žrtava fašizma 7/4 71000 Sarajevo tel. 525-212

□ Desadio vam spektrum? Odvojte 1000 din. za kasetu sa TURBO TAPE-om i MEGA bejzikom i imaćete nov kompjuter. Saša Požgaj Starih omladinaca 4 41000 Zagreb tel 041/565-158

□ Predajem komplet. DT Supertest Fighting Warrior Cokin Kennedy Approach Neverending Story A C E Fight Night, Hard Ball+kaseta (C-45(za 1100 din. Stojanović Branislav Lipa 25 11030 Beograd tel 011/558-807

□ Danas u vašem gradu svi najnoviji programi na jednom mestu Ping Pong Movie Summer Games 2 Amazon Women i puno drugih Za spisak obratite se na adresu Stevan Šavljanski Teodora Pavlovića 8 21000 Novi Sad tel 021/618-580

□ COPY DE LUX — Kopira sva spektrum programa Objedinjeni Monster i Super Copy u jeden program Jednostavan rad za početnike i za pirate. Opširno uputstvo sa primerima. Sa kasetom i ptt 1200 din Jeremić Nebojša, Risanska br 10 11000 Beograd tel 011/643-061

□ Predajem komplet 2 Hyperblaster Ping-Pong Yabba Doo West Bank, McGuigan Gladiator Komplet 3 Way of Tiger Goonies Visitors Back to Future Komplet 560 dinara. Radojković Saša, 3 oktobar 166 19210 Bor tel 030/36-182

- Samantha Fox Strip Poker Back To Future (prema filmu) Green Beret (imagine), Way of Tiger (3 programa) Ping Pong (Imagine) Bomb Jack (Elite) Pogo (Ocean) Shrinkin Fireman (Mastertronic), Amazon Woman+kazeta+poštarina+uputstvo=3000 din Šepić Miran Rastočine Š-3 51000 Rijeka, tel 051/510-855

□ SPEKTRUM SOFTWARE STUDIO — izbor od preko 1000 programa — svaki program sa uputstvom — veliki izbor literature knjige i originalnih programskih uputstava na engleskom i srpskohrvatskom jeziku Spisak programa je besplatan sa katalog sa opisom posisti 200 din Brze i kvalitetne usluge — proverite PAJNIĆ MIRKO STRAHINJIĆA BANA 56, 11000 BEOGRAD, tel 011/188-190 posle 15h

- 1500 spektrumovih programa među kojima i najnoviji superhitovi snimam na vaše kasete (30XC 90) za 16000 din Tražite katalog (sa naznakom — za megakoplete) Mario Mendeš Jeretova 5 58000 Split tel 058/553-506 iza 21 h

□ Sva najnovija programa koji se nalaze u Jugi kod nas možete jeftinije nabaviti! 11 programa po vašem izboru=800 din Snimamo i proveravamo direktno iz spektruma! Katalog besplatan! COMPUTER ART Vojvode Mišića 18 21000 Novi Sad tel 021/58-542

□ Superkomplet — 18 najnovijih spektrumovih hitova sa prevedenim uputstvima, mapama poukovima i kazetom za samo 1500 din BEST BUY SOFTWARE Mario Mendeš Jeretova 8 Split, tel 058/553-506 iza 21h

□ VRHUNSKI GRAĐEVINSKI PROGRAMI za ZX Spectrum OKVIRI REŠETKE ROŠTILJI PILOTI DIMENZIONIRANJE VODOVOD i mnogi drugi Za radne organizacije i pojedince. Besplatan katalog Gino Grecin Kozala 17 51000 RIJEKA tel 051/517-291

□ BEST BUY SOFTWARE — najnoviji programi iz Engleske Njemačke Italije i Danske u kompletima od 90 minuta za samo 500 din Violent Sex, Inferno Athlete Spitfire 40 Diller Tražite katalog Mario Mendeš, Jeretova 5 58000 Split tel 058/553-506 iza 21h

□ C jezik 1 1 Blast, Logo 1 2 Beta Basic, Learn Chess i mnogo drugih uslužnih programa sa uputstvima prodajam Tel 011/182 101

□ SERVIS ZA KOMPJUTERE ZX SPEKTRUM I C-64 Brzi i kvalitetni popravci ugradnja resete i druge usluge. Kovačić Anđelko VIII Vrbik 33a/5 41000 Zagreb tel 041/539-277

□ Na mađarskom jeziku profesionalni prevodi uputstva za spektrum Blast, Devpac, Asembler Disassembler Paint Box, SCR 2D Machine Tasword Master File Forth Beta Basic i Leonardo Tel 011/182-101

□ Spektrumovci niske cene veliki popusti najnoviji programi Rok isporuke 24 časa. Ako programe naručite telefonom 12 časova. Tražite besplatan katalog Robert Vahtler Omladinska 39 55000 Slavonski Brod tel 055/236 107

□ ROTKVA SOFT — predstavljamo vam više od 500 programa. Svi programi su snimljeni iz kompjutera i provereni Popusti stalnim kupcima. Igre april 86 V-visitors Taffy Turner F A Cup Football The Way of Tiger Bomb Jack Gman Berets Back to the Future Ite Navedene programa imamo! ROTKVA SOFT Bul bratstva i jedinstva 20/412 21000 Novi Sad tel 021/398-454

□ TEST INTELIGENCIJE — kroz 180 pitanja dobićete potpunu sliku o svom koeficijentu inteligencije Cena programa 800 din Besmrtni POKE katalog — 260 poukova za 130 igara za spektrum Pošaljite preporučeno 400 din Tot Tiber B Dejanović 1 21250 Sr Karlovci tel 021/881-650

□ Spektrum — najnoviji i najbolji programi u kompletima Komplet 54 Turbo Espirit Spellbound Jetset Willy 4 Forbidden Planet Mugsys Revenge Ping Pong Spitfire 40 Amazon Women V-visitors The Way of Tiger 1—4 Komplet 53 Battle of Planets Championship Boxing International Rugby Movie Code Name Mat II West Bank Cylu Video Olympics Rapecallion Frankenstein 20000 Sky Fox Yabba Dabba Doo Jeden komplet+kaseta+PTT 1500 din Trbca Goran Stevana Lukovića 9 11000 Beograd tel 011/563-348

□ MACHINE CODE FOR BEGINNERS — sada i Vama data efikasnost mašinskog programiranja. Traka i opširno uputstvo (65 str) — 2000 din VRCA MILAN Zarija Vujoševića 79 11070 Novi Beograd

□ BLAST! ! ! — ubrzaće Vaše bejzik programe i do 40 puta (vidi RAČUNARE 14 strana 25) Traka i opširno uputstvo (35 str) — 2000 din VRCA MILAN Zarija Vujoševića 79 11070 Novi Beograd

□ SPEKTRUMOVCI! Novoformirani Kermitsoft od sada zauvek sa vama! najnoviji programi stalno stižu iz Engleske Još uvek snimamo programe direktno iz računara i vršimo verifikaciju svakog programa. Požurite! Odavno imamo Ping Pong Runestone V-visitors Gman Beret, Knight Rider Way of Tiger Dybe run Bomb Jack Tražite besplatan spisak na adresu Jerković Branko Gročanska 47 11000 Beograd III na tel 011/428-358

□ Spektrum — 12 programa za učenje engleskog+kaseta 1000 din 25 radioamaterskih programa+kaseta 1000 din 40 COPY programa+kaseta 1000 din. Tražite besplatan katalog sa preko 1000 programa. Trbica Goran Stevana Lukovića 9 11000 Beograd tel 011/563-348

□ DYNAMIC PROGRAMMING — specijalni efekti i više od 200 novih komandi u bejziku Traka i opširno uputstvo (27 str) — 2000 din VRCA MILAN Zarija Vujoševića 79 11070 Novi Beograd

□ HAPPY SOFT nudi najnovije najjeftinije programe i literaturu za komodor Direktno sa engleskih top lista. Nagrade popusti besplatan katalog Victor Karaba Braća Drobnjak 17/50 21000 Novi Sad tel 021/396-257

□ BLAST! ! ! DYNAMIC PROGRAMMING MACHINE CODE FOR BEGINNERS trake i uputstvo pojedinačno — 2000 za dva programa — 3000 sva tri — 4000 din VRCA MILAN Zarije Vujoševića 79 11070 Novi Beograd

□ Spektrumovci vama Kaj-soft nudi komplet K-2 Beach Head 3 Rock N Wrestle Ping-Pong Tau Ceti Cyberun International Rugby Back To Future Dynasty Komplet+kaseta+PTT=1000 dinara. Kajsoft Diljska 20 54000 Osijek

□ Kaj-soft nudi spektrumovcima komplet K-3 Movie Battle of Planets Goonies The Way Of Tiger Video Olympics, Zoids Westbank Barry McGuigan Thunderbirds Amazon Women Strong Man Cuberun Kvalitete je naš moto Komplet+kaseta+PTT=1000 dinara Kaj soft Diljska 20, 54000 Osijek

□ COMPUTER STUDIO nudi
Komplet 53 BC Quest Zorro Jet Set Willy 3
Komplet 54 Gunfright Winter Games Tau Ceti
Uz to mnogo programa, literatura hardverski dodaci saveti pomoć
Četiri godine iskustva, nepovećavanje cene od 1983 preko 10 000 zadovoljnih mušterija iz zemlje i inostranstva govore nama u prilog
COMPUTER STUDIO Starec Bojan Kosančićev Venac 1A 11000 Beograd tel 011/625-833

□ COMPUTER STUDIO nudi
Komplet 55 Pyjamarama 4 Beach Head 2 Jet Set Willy 4 Boulder Dash 2
Komplet 56 nastavak Hobbita, Thunder-Birds, International Rugby
Uz to mnogo programa literature hardverski dodaci saveti pomoć
Četiri godine iskustva nepovećavanja cene od 1983 preko 10 000 zadovoljnih mušterija iz zemlje i inostranstva govore nama u prilog
COMPUTER STUDIO Starec Bojan Kosančićev Venac 1A 11000 Beograd tel 011/625-833

□ SUN SOFTWARE CLUB — spektrum je tri godine sa vama, zahvaljujući kvalitetu i bogatom izboru više od 1500 programa. Programi su podeljeni u nekoliko grupa uslužni sistemski copy kao i najnovije igre Pri naručivanju svi programi su snimljeni direktno iz računara a svaki pojedinačno i proveren Sva najnovije igre čije nazive možete naći u drugim oglasima a koje su trenutno u Jugoslaviji već posedujemo Naročite povoljnosti uživaju članovi i stalni kupci Posebni popusti za veće narudžbine Kvalitetno i brzo! Veliki katalog na 15 strana je 200 din a mali katalog programa je besplatan Majerovec Ivan Vojvode Mišića 2/5 21000 Novi Sad tel 021/57-988

□ Veliki izbor kvalitetno pojedinačno snimljenih spektrum programa. Za besplatan katalog javite se na 011/436-137 do 15 časova. Miroslav Radosavljević Braće Nedića 2 11000 Beograd

□ COMPUTER STUDIO nudi
Komplet 57! Blade Runner Spitfire Adrian Mole
Komplet 58 Turbo S Print, Viking Yabba Dabba Doo
Uz to mnogo programa literatura hardverski dodaci saveti pomoć
Četiri godine iskustva nepovećane cene od 1983. preko 10.000 zadovoljnih mušterija iz zemlje i inostranstva govore nama u prilog
COMPUTER STUDIO Starec Bojan Kosančićev Venac 1A 11000 Beograd tel 011/625-833

□ Spektrumovci ! ! ! Komplet 21 YABBA DOO SPELLBOUND DUSTMAN TURBO ESPRIT MUGSY REVENGE INFERNO Alien SWARM ATHLETE PAR POSTMAN TUNELES DISASTER STARSHIP ENTERPRISE GALACTIC AUTOSTOPER RETUR OF THINFS Komplet 22 REAL YOU1 2 ADRIAN MOLE 1—4 SKY FOX JET SET WILLY4 FRANKENSTEIN2000 CODE NAME MAT 2 MEGA FRUIT CHOPPERX—1 SWORDS AND SORCERY Cena jednog komplete 800 din Radović Branislav Sonje Marinković 14/4 21000 Novi Sad tel 021/28-682 ili 022/424-824 (vikendom)

□ Mc SOFTWARE! SPEKTRUMOVCI!
Hitovi aprila 1986 u kompletima od 14 programa za 800 din +kaseta. Rok isporuke 1 dan KOMPLET 32 Pyjamarama 4 (Three Weeks In Paradise) Winter Games 1 2 (US Gold) Thunderbirds Riddlers dan The Arc Of Yesod Stmal Hawk (Ocean) Gladiator Grumphy Grumphley Gremlins, Zoids Strong Man Culy. International Rugby KOMPLET 31 Zorro (US Gold) Gunfright (Ultimate) XCEL Wham — The Music Box Mr Freeze Rockman Jason Gem 2112AD Tauceti Juggernaut Astro-Clone Panzadrome Zenji Starquake (Wizards Lair 2) KOMPLET 30 Elite Nomad Mikie Transformers (Ocean) SW World Sub Striker Sir Fred Metabolis Enigma force Cos. Wartoad Hyperblaster Think, Bimbo Tombola Zoran Milošević, Pere Todorovića 10/38, 11030 Beograd tel 011/552-895

□ Spektrumovci direktno iz Londona ! ! ! Komplet 23 AMAZON WOMEN PING PONG McKENSIE TITANIC FRIDAY 13 FINEST HOUR RUPERT RUNE STONE ASHKERON CANNON BALL, OVERLORDS FUZZIE WUZZIES ALL OR NOTHING SAM SPADE Komplet 24

WAY OF TICAP 1—3 GREEN BARET V Visitors F.A FOOTBALL TAFFY TURNER BOMB JACK UNCREDIBLE FIREMAN SOUND MASTER XENO2 AUSTERLITZ, YUSKOOL DAZE Cene jednog kompleta 800 din Radović Branislav Sonje Marinković 14/4 21000 Novi Sad tel 022/28-682 ili 022/424-824 (vikendom)

KOMODOR

○ KRAPINKO SOFTWARE — najnoviji hitovi po najnižim cenama za komodor 64 Programi u kompletima i pojedinačno Rok isporuke 48 sati Katalog besplatan Kiralj Denis Marka Orešković 1/7 55000 Slavonski Brod tel 055/238-886

○ Komodror 64 — paket programa Nexus, Neverending Story Rave Kane Ping-Pong, Enigma Force Staff 6+kaseta=1800 din Moguća razmjena najnovijih programa Lasić Hrvoje Mike Katunara 8 51000 Rijeka tel 051/442-856

○ KOMODOR HARDWARE — Novo iz P N P electronica. Dodaci sad i za komodor 64 ROM moduli sa mnogo dobrih programa. Nema više upisivanja sa kasetofona, jer modul postaje sastavni dio kompjutera. Laji pen kabel sa monitor EPROM programator Sentroniks printer interfejs i još mnogo toga POPRAVCI Besplatni katalog P N P electronic, Jeretova 12 58000 Split

○ TD SOFT — sve za C-64 na jednom mestu Pišite nazovite i tražite besplatan katalog Marković Tomislav Pačirski put 8 24000 Subotica tel 024/28-888

○ Izuzetna prilika za komodor 64! Prodajem 80 vrhunskih igara za samo 2000 din Kaseta besplatna. Prvih deset naručilaca dobijaju po 10 besplatnih programa na istoj kaseti Ilić Mališa, 12307 Tabanovac

○ NOVO ZA KOMODOR 64!!! Profesionalni program za obradu teksta srpskohrvatski karakteri mogućnost štampanja svih grafičkih znakova, (lako konstruisanje tabele itd) mogućnost definisanja 30 proizvoljnih karaktera kompatibilnost sa Easy Script-om Sve ovo za printere MPS 801 802 803 i VIC 1526 Dish tivadar Bolmenzka 96 24220 Čantavir

○ Komodor 64 — stalan priliv najnovijih programa za kasete u paketima ili pojedinačno. Informacija na tel 011/102-914

○ Komodor 64 — komplet A Robin of the Wood Last V8 Yabba Dabba DOO Mikie Karateka, Podešavanje glave Ping-Pong Staff 2 3,4 5, Imhotep Yie Ar Kung Fu Kawasaki William Wobbler Komplet B Kawasaki RR Yie Ar Kung Fu Who Dares Wins 3 Commando 1 i 2 Back To Skool Dragon Skull Mikie Staff 2 3 i 4 ACE The Young Ones Komplet+kaseta=1800 din Svi programi su u TURBO TAPE-u Ninić Goran Tomislavova 3 43290 Križevci tel 043/841-870

OBRADUJTE SVOJ RAČUNAR! OBJAVITE MU MALI OGLAS U „RAČUNARIMA"!

○ DALMATIA SOFTWARE vam predstavlja najbolje i najjeftinije programe za vaš Komodor 64 ELITE, OXFORD PASCAL PIZZA 202 RAMBO II i još mnogo toga. Tražite besplatan katalog DALMATIA SOFTWARE Skojevaca 4/VI 58300 Makarska tel 058/613-891

○ Komodor 64 — prodajem najnovije kasetne programe Monty Nodes Of Yesod F T Supertest, Elite 2 Back To Future Treasure Island 2 Staff 4 i 5, Commando 2 Robin Of the Wood Desert Fox, Alien 8 trodimenzionalan i ostale Hajdina Damir Štajnbergerova 42 41040 Zagreb tel 041/253-338

commodore
za sva vremena

BASIC
i
Simon s BASIC

Jasno pregledno i potpuno izložen je osnovni jezik Komodora 64 sa preko 150 primera u knjizi COMMODORE ZA SVA VREMENA. I još mnogo toga!

336 strana 16 x 23 cm latinica
Cena 3600 din
Način nabavke U svim knjižarama ili direktno od izdavača
Mikro knjiga
P O Box 75 11080 Rakovica — Beograd

○ Prodajem najbolje najnovije programe za C-116 Grandmaster Lawn Tennis Big Attack, Canoe Slalom Olympic Skier Iznenađenja. Poklon programi Mitrović Vukašin Vojislava Ilića 90 11000 Beograd tel 011/4898-168

○ EKSKLUZIVNO ZA C-64!!! Ako želite najbolje najjeftinije (30 d) najkvalitetnije presnimljene programe obratite se BIRC-SOFTU Na 10 naručenih programa dobivate 1 besplatno Katalog je besplatan Grobenski Milivoj — Birc F Stareje 10 42000 Varaždin

○ Komodor 64=hitovi maja Run For Gold Comic Bakey Shoping, Yie Ar Kung Fu Mercenary Who Dares Wins 3 Soccer 4 sa kasetom 1200 din Pojedinačno Ping-Pong Bounder Rock n Wrestle Svi programi u kompletu su TURBO Dejan Grubar Stojana Jankovića 64 11000 Beograd tel 011/581-516

○ L — SOFT Mnogo novih programa za komodor 64 Besplatan katalog sa najjednostavnijim načinom naručivanja Levak Nenad Kumičićeva 14 42000 Varaždin, tel 042/40-603

Komodor 64 — prodajem i menjam najnovije programe Najveći hitovi u kompletu Sky Fox Mervenary Neverending Story DT Supertest USA Road Race Imhotep, Fight Night Gramlins Komplet+kaseta+poštarina 900 din Možete poručiti i neke druge programe Katalog besplatan. Teofilović Dragoljub Vojvode Brana 7 11000 Beograd tel 011/422-706 ili 752-783 Aca

○ Komodor 64 — poslovni programi
— EASY SCRIPT — profesionalna obrada teksta
— WISAWRITE — prof obrada teksta (datoteke samo na disku)
— PRACTICALC — obrada i sortiranje tabeli raznih podataka 22 matematičke operacije itd
— Statistički paket programa
— HELP 64 — nezamenljiv za ozbiljna programiranja Svi programi osim pod 4 su sa kvalitetnim srpskohrvatskim uputstvom Jedan program 1800 din svaki sledeći 1300 din Milenović Mladen Lenjinova 2C 71000 Sarajevo tel 071/210-834

○ Boulder Dash 4 Yabba Dabba Doo Superman 2 Turbo Pizza 202 Space Pilot 2 3 American Road Racer Back To The Future komplet+Kaseta=1360 din Moguća i razmena. Perunović Vladimir blok 8 Al ulaz i 81000 Titograd tel 081/11-379

○ Komodor 64 — preko 1700 prošlih sadašnjih i budućih hitova koje možete nabaviti pojedinačno ili u kompletima uz nic popusta i niske cene Komplet R GOONIES FOURTH PROTOCOL LITTLE COMPUTER PEOPLE PING PONG ELITE II (Mercenary) MISTICAL MISSION GYROSCOPE II TIME TUNNEL AMAZON WOMEN PRO BOXING TOMMY AND JERRY SPIDER THE FLY Komplet S DORIATH SATANS GROTTY CHRISTAMS UHG TALES OF THE CAT COMPANU AND CO WILLY TEX LABIRINTO INTHERTIANCE TRIBLE JUMP TAXI METRANAUT IRIDIUM ROCK'N WRESTLE Komplet T STRONG MAN ELECTRA GLIDE SONG ELECTRA, TROOPER CROUPER TIPO CAMELEON, NUDIST CAMP THE BLACK CELEBRATION LITTLE COMPUTER PEOPLE SABOTEUR, POLE POSITION II CASTLE WOLWENSTAIN II Svaki komplet — 1500 din Komplet svih 1700 programa (uključujući i gore navedene) možete dobiti po ceni od samo 200000 din +kasete (vaše ili moje) Detaljna obaveštenje tražite na tel 015/25-772 ili na adresu Branko Vrhovac Moše Pijade 4 15000 Šabac

○ Komodor 20 16 +4 64 128 programi Tražite besplatan katalog Derman Šandor Rade Končara 23 23000 Zrenjanin
RTTY CW konačno i za C-64 Informacija na adresi Kovač Jani Klavnička 13 69000 Murska Sobota

○ Turbo P S — novi programi za vaš C-64 korisnički programi igre Copy programi upute i saveti za početnike Katalog besplatan. Koštrun Marijan Try I Kukuljevića 14, 41000 Zagreb

○ Rasprodaja! Komodorci jedan ogroman paket+kaseta sve hitovi staje samo 999 din Može i posebno. Nazovite ili pišite 021/812-309 Grozdić Dalibor Žarka Zrenjanina 3 21220 Bečej

○ Komodor 64 — komplet 40 najboljih korisničkih programa sa svih područja +kasete+poštarina=3999 din! Pogodno za škole radne organizacije i pojedince! Prodaja i zamena najboljih igara! Kratin Zoran, Prvomajska 2 69000 Murska Sobota tel 069/24-526 (posepodne)

○ Komodor 64 — komplet od 100 igara za samo 1000 din Besplatan spisak Exploding Fist, Commando Skol Daze Brza i kvalitetna isporuka. Sajčić Đorđe Boška Savića 7 37240 Trstenik

○ Komodor 64 komplet 1 Winter Games Petak 13 Elite Aligata Superman Rambo 2 Komplet 2 Commando Rocky Horror Show Rambo 2 Robin of Wood Elite Who Dares Wins 2 Komplet+kaseta 1000 din ANE SOFT Kostурska 77 91400 Titov Veles tel 093/25-387 20-334

○ HAPPY SOFT nudi najnovije, najjeftinije programe i literaturu za komodor Direktno sa engleskih top lista. Nagrade popusti besplatan katalog Victor Karabe. Braće Dronjak 17/50 21000 Novi Sad tel 021/395-257

○ Najnovije igre za C-64 u kompletu. Komplet A Robin of Wood Yie Ar Kung Fu Rally Afric Madonna Falkland's Thunderbirds Bounced (neviđena berba) Touch Foozball (regbi) Castle Of Creet, Hypersport 2 Cena kompleta sa kasetom iznosi 1500 din Programe možete naručiti na adresu Petrović Mir— Jovana Stojisavljevića 9/11 11080 Zemun tel 011/195-758

commodore
za sva vremena

MAŠINSKO PROGRAMIRANJE MAPA MEMORIJE I ROM RUTINE

Kompletan kurs mašinskog programiranja sa detaljnim objašnjenjem svake memorijske lokacije daje u knjizi COMMODORE ZA SVA VREMENA I još mnogo toga!

336 strana, 16 x 23 cm latinica
Cena 3600 din
Način nabavke U svim knjižarama ili direktno od izdavača
Mikro knjiga
P O Box 75 11080 Rakovica Beograd

□ Komodor 64 — najnoviji sportski programi Winter Games (5 Igara)+American Road Race+speed King+International Basketball+Slapshot Hockey+McGugan Champion Boxing+kaseta+poštarina=1300 din. Plaćanje pouzećem Predrag Sljepčević, Bulevar revolucije 290/V 11000 Beograd tel 011/417-064

□ ROMANTIC ROBOT SOFT — programi za C-64 Svi razbijeni Pristupačne cene besplatan katalog SCOTCH kasete. COPY PROGRAMI — NPR Turbo 250 Uz svaku narudžbu Turbo 1002 Spiskovi najboljih u Računarima 11 12 13 14 najnovije Human Race Dynamite Dan Kapriolan McGuigans Box, Enigma Force Nodes Of Yesod Black Wuche Wizardy Fight Night Underwurlde Svaki mjesec i do 30 novih hitova Žuljević Kemal Braće Kosorića 13 72220 Zavidovići tel 072/574-441

□ Komodor 64 za sva ljubitelje muzike komplet najboljih muzičkih programa (Funkyy Drummar Synth Sample ...) 7 programa+kaseta+poštarina=1000 din Plaćanje pouzećem Predrag Stjepčević Bulevar revolucije 290 11000 Beograd tel 011/417-064

□ Za komodor 64 VIC 20 C 128 prodajem konektora (priključke) za kasetofon user-port džojstik te produžni kabl za džojstik reset taster Cijena pojedinačnog programa po izboru za C-64 samo 25 do 35 dinara Jeftina literatura i uputstva za programe Razmjena programa i literatura Besplatan katalog Odgovaram svima Tel 074/832-832 Radovanović Milorad Radojke Lakić 3 74400 Derventa

□ Komodor 16 116 +4 Veliki izbor programa Novi hitovi Bejzbol World Cup Fudbal Berks Rockman katalog besplatno Ljubisavljević Dragan 3 oktobar 302/9 19210 Bor tel 030/33-941

□ SERVIS ZA KOMPJUTERE ZX SPEKTRUM I C-64 Brzi i kvalitetni popravci ugradnja reseta i druge usluge Anđelko Kovačić VIII Vrbik 33a/8 41000 Zagreb tel 041/539-277

□ KOMODOR SOFT — biblioteka od oko 2000 programa Tražite besplatan katalog Bićete ugodno iznenađeni Aušperger Nada Vladimira Gortana 86 42000 Varaždin

□ KOMODORCI Najjeftinija i najbolja ponuda hardvera i softvera Hardver uređaj za direktno presnimavanje (2000) reset (800) i mnogo drugih uređaja po vašim i našim šemama. Softver Fight Night Back to Future Commando 3 Night Shade DT Supertest Neverending Story Staff 5 The Rats Funky Drummer Karateka 3D+kaseta=1800 dinara i mnogo pojedinačnih programa Lazić Miroslav Slavka Rodića 14-b 72000 Zenica tel 072/23-963 i 072/38-131

□ JOKER SOFWARE nudi za vaš C-64 Little computer peple Transformers Dragon Skull Robin of the Wood Dna Man Bans Sys Droid Naručite besplatan katalog na tel 021/396 245 Jocić Dragan Put bačkopalanačkog odreda 20/2 stan 21 21000 Novi Sad

□ Komodor 64 Back to Future 2 Enigma Forrce 2 Red Arrows Who Dare Run for Gold Yie Ar Kung Fu Soccer 4 Flight Simulator 3 Soho Jumper Programi+kaseta 1300 din Pojedinačno Supertest Elite 2 ... Mladen Jevtić Crvenih hrastova 16/7 11030 Beograd tel 011/544-428

commodore

HARDVER
I
KONSTRUKCIJE

Potpuna električna šema Komodora 64 sa objašnjenjima načina rada i uputstvima za gradnju interfejsa modema EPROM programatora i kartridža data je u knjizi COMMODORE ZA SVA VREMENA i još mnogo toga!

338 strana 18×23cm latinica
Cena 3600 din
Način nabavke U svim knjižarama ili direktno od izdavača
Mikro knjiga
P O Box 75 11090 Rakovica Beograd

□ CBM 64 IMPOSSIBLE AND MIRACLE SOFT predstavljaju vam komplet igara čije su recenzije upravo izašle u engleskim časopisima Komplet 22 Karateka 2 Mercenary 3 Elite (sa celokupnom opremom+30 hiljada kredita) Little Computer People Nexus Boulderdash, Zorro Kung Fu Master Nude girl (5 golih devojaka iz strip pokera) Komplet možete naručiti na adresu Nenad Skendžić Lenjinova 85/II 21205 Sremski Karlovci tel 021/881-908 (Robert Ranković)

□ DX EUCALIPTUS vraća se sa najnovijim hitovima i super ludim paketima (Yie Ar Kung Fu Captive The Rats ...) Sve to u besplatnom katalogu Davčić Zoran Nikole Tesle 12 51440 Poreč tel 053/33-836

□ Izmena na štampačima KOMODOR 1526 i MPS 802 uz prateće programe omogućiće vašem KOMODORU 64/128
— kreiranje sopstvenih znakova i kompletnih setova slova
— proizvoljan raspored na tastaturi
— pisanje i štampanje teksta sa izborom seta po želji (ćirilica nemački matematički simboli itd.)
— hardcopy ekrana visoke rezolucije iz SIMON's BASICa
Tel 011/331-306 i 426-422

□ BESMRTNI POKE KATALOG — 280 poukova za 138 igara. Komodorovci otkrijte tajnu omiljenih igara Za katalog poslati preporučeno 400 din Tot Tibor B Dejanović 1 Sr Karlovci 21205, tel 021/881-650

□ Komodor 64 — najnoviji programi za samo 50 din (Nodes of Yesod Jorth Protocol Goonies) i mnogo drugih Besplatan katalog Moguća razmena Šafranić Alan Hrašće 20e 41020 Novi Zagreb

□ Za vaš komodor prodajem paket od 15 programa za 800 din i to Beach Head 2 Match Point Zorro Hyper Sports Fist Basketball 2 Olympics ... Mojsin Zoran Al spomenica 5/38 19210 Bor tel 030/25-882

□ Spektrum — profesionalni prevodi Napredni mašinski jezik 1800 d ROM disassembley 1500 d Mašinski jezik za apsolutne početnike 1300 d Spektrum priručnik 800 d 50 tajni bejzik programiranja 800 d Devpac 3600 d Devpac 3 na kaseti 800 d Mega bejzik 500 d Mega bejzik na kaseti 800 d Tražite besplatan katalog Trtica Goran Stevana Lukovića 9 11000 Beograd tel 011/563-348

□ Komodor 64 — komplet od 15 programa+kaseta+poštarina+pokloni=1000 din Programi su Commando Elite Hacker Pyjamarama 1 2 1 3 Boulderdash 1 2 1 3 Skool Daze Kremenko Popović Tihomir Vojvode Stepe 10 32000 Čačak, tel 032/33-69

□ Komodor prodajem komplet: Hyper sports Hyper Olympics Hacker Exploding Fist, International Basketball 2 Squash McGuigan Robin Beachhead 2 ... i još 10 igara za 450 din Radojković Saša 3 oktobar 188, 19210 Bor tel 030/38-192

□ PIZZA SOFT komodor programi Commando Exploding Fist Beach Head 2 Elite Hacker Zorro Rambo 2 Cene niske. Isporuka brza Besplatan katalog Rusovan Milan Koste Racina 11 21000 Novi Sad tel 021/316-885

□ Komodor 64 — profesionalni prevodi Reference Guide 1700 d Mašinski jezik 1300 d Mašinski jezik za početnika 1400 d Priručnik od C-64 1300 d Kako da programirate C-64 800 d Disk sistemi i štampači 900 d Grafika i zvuk good Matematika na C-64 1000 d Simons basic 700 d C-64 basic 700 d Practicalc 800 d Easy Script 400 d Multidata 400 d Vizawrite 400 d Pascal 400 d Help 400 d Graf 400 d Isporuka za 24 časa. Trtica Goran Stevana Lukovića 9 11000 Beograd tel 011/563-348

□ COMMODORE 64 i ovog meseca komplet najnovijih programa Komplet A FOURTH PROTOCOL GOONIES STAFF 5 NEVERENDING STORY NODES 2 DT SUPERTEST NIGHT SHADE SPACE PILOT 3 RATS KARATEKA Komplet B WORLD CUP 2 SABOTEUR LAST V8 TOCH FOOTBALL YIE AIR KUNG FU CASTLE WOLFENSTEIN MERCENERY FARM SONG COMIC BAKERY RUN FOR GOLD Komplet+kaseta=1500 din Katalog besplatan Milan Stamenović ul Vladimira Nazora 77 18300 Pirot tel 010/24-382

NOVO!



NOVO! NOVO! NOVO!
TROJELNA UTICNICA
RJEŠAVA PRIKLJUČAK VIŠE MALIH POTROŠAČA SA PLOSNATIM UTIKAČEM SNAGE UREĐAJA DO 3x500 W. NPR TV VIDEO HI FI LINIJE GRAMOFON, KOMPJUTORI- SUŠILO ZA KOSU BRIJAĆI APARAT, FIGARO ITD CIJENA 450 D/KOM

TOMISLAV ŽIVKOVIĆ
41000 ZAGREB
BARČIĆEVA 5/2

trodjelna utičnica

□ KING'S CLUB vam nudi najbolje i NAJJEFTINIJE programe za C-64 ELITE AMERICAN ROAD RACE MC GUIGAN BOXING SPECTRUM SIMULATOR i mnoge druge. Katalog besplatan. Denis Paliska, Slavića Vajnera Čiče 9 51000 Rijeka tel 061/31-025

□ MPS-802 paket za profesionalnu obradu teksta. EPROM sa latinicom i prilagođeni program Kesler Viktor Rumenačka 106-1 21000 Novi Sad tel 021/334-717

□ Razdelnik DVATASET za priključivanje dva DATASETA i presnimavanje programa. Novo DVATASET+presnimavanje u oba pravca verifikacija ugrađen zvučnik dva preklopnika solidna konstrukcija. Kesler Viktor Rumenačka 106l 21000 Novi Sad tel 021/334-717

AMSTRAD

□ Amstrad — profesionalni prevod Priručnik CPC 464 1300 d Mašinski jezik za početnika 1300 d Locomotiv basic 1300 d Devpac 700 d Master file 700 d Pascal 700 d Tasword 400 d Isporuka za 24 časa. Trtica Goran Stevana Lukovića 9 11000 Beograd tel 011/563-348

□ Komplet vrhunskih programa za amstrad Formula 1 Arabian Nights Relly 2 Match Point Dam Barch Project Future Finders Keepers Air Wolf 3D Grand Prix Beach Head Kazeta+poštarina 1400 din Mann Fulgosi Savska 9 41000 Zagreb tel 041/442-056

□ CAPTAIN SOFTWARE — najnoviji programi za amstrad CPC 464 Pojedinačno i kompleti uz niske cene Tražite besplaten katalog Captain Software Margaretska 3 41000 Zagreb tel 041/438-087 od 18—20h

□ Za amstrad prodajem paket od 10 programa za 2500 din i to Winter Games Zorro D T Supertest Yie AR Kung Fu Bruce Lee ... Mojsin Zoran, Al Spomenice 5/38 19210 Bor tel 030/25-882

□ AMSOFT YU predstavlja najnovije CP/M programe Jedino kod nas možete nabaviti Prolog Lisp Cobol Algol Wordstar 3 30 Datastar Multiplan, C-kompajler Turbo Pascal Najnovije igre 3D Time Trek, The Key Paccor The Frize Sorcery plus Highway Encounter Myrodina flight Simulator Posebna pogodnost kompleti programa na kazetama ili disketama (10—20 programa na kazeti 2800 din) Velika ponuda ostalih uslužnih programa na kazeti ili disketi AMSOFT YU Trg republike 4 41000 Zagreb tel 041/315-478 ili 270-777

□ Amstradovci veliki izbor jeftinih programa Jedan komplet+kaseta 1400 din Može i pojedinačno Tražite besplatan katalog Matić Vladen Ive Lole Ribara 38 11319 Mločevac

□ Kupujem programe CW SSTV STV RTTY i sve iz radioamaterske oblasti za CPC 464 Ponude slati na adresu Lazar Paralić Narodnog Fronta 14 21000 Novi Sad

ATARI

□ EMPELE ATARI CLUB — najnoviji i najbolji programi za atari 600/800 XL Jedan bejzik program 100 din a mašinski program 200 din Igor Petraševski Emil Zola 7/23 91000 Skopje tel 091/216-478

o ATARI ST 520/+ kupujem razmenju jem prodajem programe i literaturu Nenad tel 011/673-733

HARDVER

o Prodajem spektrum 48 K sa tastaturom TREND originalnom spektrum tastaturom, kempston interfejsom džojstikom pratećom literaturom i 25 kaseta sa programima. Može i pojedinačno Prednost imaju kupci iz Sarajeva. Markušević Snežana Lenjinova 87/IV 71000 Sarajevo, tel 071/612-438

o Prodajem galaksiju 6/6K sa originalnom kutijom i ispravljačem Vidaković Dražen Nodilova 5 41000 Zagreb tel 041/416-120

o Prodajem komplet čipova za galaksiju 6+6K memorijsko proširenje 32K (sa konektorom) programator EPROM-a sa Textool podnožjem i konektorom najboljem ponuđaču Berberski Žarko Dranovačka 2 11000 Beograd tel 011/772-203

o Prodajem komodor 64+dve palice+kasetofon+50 najnovijih programa za 12 starih miliona. Milojević Ivan Staro naselje 8 38218 Leposavić

o Prodajem amstrad 6128 sa zelenim monitorom i dodatnom disk jedinicom i komodor 128 sa flopi diskom 1571 Novo i ocarinjeno Tel 011/699-011

o SERVIS ZA KOMPJUTERE C-64 I ZX SPEKTRUM Brzi i kvalitetni popravci ugradnje reseta i druge usluge Kovačić Anđelko VIII Vrbik 33a/8 41000 Zagreb, tel 041/530-277

o Prodajem spektrum 48K, verzija 4B sa profesionalnom tastaturom reset prekidačem zvukom na TV video-monitor izlazom ZX printer programabilni kalkulator TI-58 Tel 021/27-530 od 20 do 22 časa

o ACORN BBC B prodajem Standardni ISO-Pascal (končen u Radionici logičkih igara) mikro-Prolog asembler i BASIC — svi u ROM u monohromatski monitor Philips svi kablovi kompletna originalna dokumentacija i svi priručnici za jezike sve na engleskom Samo zajedno 350 000 din Jurković Rozmanova 13, 61000 Ljubljana

o Prodajem kompjuter Spektra Video 328 sa 32K ROM i 80K RAM Micro Soft BASIC profesionalna tastatura. Uz kompjuter prodajem kasetofon kertridž kasete sa originalnim programima i literaturu Čeda Dimitrijević, Medakovićeva 58 stan 62 11000 Beograd tel 011/468-270

o Prodajem rubni konektor za spektrum (2800 din) komplet za proširenje RAM-a spektruma sa 16K na 48K (ili 80K 11000 din) te čipove 4116 4164 (1200 d) 4416 (4000 d) Z80A CPU Z80A PIO 8255 6116 2716 2732 (2400 d) 2764 (2800 d) 27128 (3500 d) AY 38910 AY 38912 (5800 d) 6502 6522 (4600 d) 1488 i 489 (1200 d) LM 1889 (4200 d) ABC 0804 ZN 427 ZN 428 (5800 d) TEXTOOL 28 (5900 d) i ostale CMOS i TTL čipove Brzo i kvalitetno servisiram računare spektrum i komodor (C-64 C-16 C-116 C-4) N Ćetković J Leskovara 1 42000 Varaždin tel 042/38-58

o Matrični printer Seikoša GP 50s sa interfejsom za spektrum prodajem ili menjam za mikrodrajv Petković Branislav Jurija Gagarina 122/44 11070 Novi Beograd tel 011/160-694

o Prodajem amstrad 6128 sa zelenim monitorom i QL sa 64K RAM s Proširenje memorije od 512K za QL novo ocarinjeno Miodrag Drenovčanin Lenjinova 1/II 22320 Inđija, tel 022/52-943

o SPECTRUM HARDWARE — KEMPSTON INTERFEJS (KEMPY) REDOSTIK PALICA (REDY) CIJENA KOMPLETA 15000 din EPROM programer Megarom P N P ROM (prapravljeni ROM) laj pen Centronika interfejs I/O port A/D i D/A konvertori jednostruki i dvostruki interfejsi za palice audio pojačalo izlaz za monitor literatura, usluge printera programiranje EPROM-a, savjeti i POPRAVCI Besplatan katalog na adresu P N P electronic Jerelova 12 58000 Split

o Prodajem monohromatski (zeleni) monitor Filips (20 MHz 820 linija) tonski 40/80 znakova, neraspakovan ocarinjen Kvikšot II i diskete 5,25 Tel 011/434-695

o Prodajem Ejkorn elektron povoljno Tel 011/328-957

o Prodajem Alphacome 32 printer za ZX spektrum sa jednom rolom termalno papira Ivčević Zdravko Vlaška 83 41000 Zagreb tel 041/445-546 posle 15h

o Prodajem komodor 64 kasetofon palice Turbo Tape kertridž ili menjam za Casio PB 770 ili Sharp PC 1500 Jovanović Aleksandar Georgija Jakšića 13, 15300 Loznica tel 015/82-406

o Prodajem komodor 64 kasetofon džojstik literaturu i programe. Omerbašić Dževad S Karića 14 74470 Odžak tel 074/770-592

o Prodajem računar galaksija 6K ROM 6K RAM kvalitetno urađen u tvorničkoj kutiji Cijena 35000 din Mandarić Branko Kralja Tomislava 47 54400 Đakovo tel 054/844-756

o Komodor 64 128 — uštedite novac! Umesto skupog komodorovog kasetofona kupite original interfejs za svaki obični kasetofon Urađen u minijaturnoj kutiji sa kablom priključcima i uputstvom 3600 din. Profesionalni kvalitet! Garancija jedne godine Slobodan Šoškić Bulevar 23 oktobra 87 21000 Novi Sad tel 021/59-573

o Komodor 64 128 — naš masterfork omogućava priključenje dva kasetofona istovremeno? Presnimavanje svih i zaštićenih programa 70% brže! Moguće kopiranje cele kasete odmah! 2600 din. sa uputstvom Garancija jedna godina! Slobodan Šoškić Bulevar 23 oktobra 67 21000 Novi Sad tel 021/59 573

o Povoljno prodajem VIC-20 sa programima Tel 018/52 708 posle 16h

BBC

o BBC-ELEKTRON! Veliki izbor ROM-ova idealno za RAM proširenja. Pavlović Ilija Vojvode Stepe 293/38 11000 Beograd

prijem malih oglasa za broj 16 zaključujemo 22. maja u 12 časova

Literatura

o KOMODOR 64 — najkvalitetniji profesionalni prevodi Programmers Reference Guide — 1750 d Mašinski jezik za početnike — 1550 d Grafika i zvuk na C-64 Umjetnost grafike na C-64 Basic priručnik Simons basic — sve po 1150 d i Pascal — 800 din Na višestruke narudžbe popust 10%! Duško Bjelotomić Centar 1 54550 Valpovo tel 054/82-665 ili 041/683-141

o AMSTRAD CPC 464 — odabrani profesionalni prevodi Uputstvo za rad na amstradu — 1650 d Locomotive basic — 1550 d Mašinsko programiranje za početnike — 1450 d Uputstva za Devpac i Tasword po 1350 d Grafika i zvuk na CPC 464 — 1350 d Na višestruke narudžbe popust 10%! Duško Bjelotomić Centar 1 54550 Valpovo tel 054/82-665 ili 041/683 141

"KOMPJUTER BIBLIOTEKA"

vam predstavlja novu biblioteku kompletnih uputstava za uslužne programe:

1. NOROSTAR CP/M 3.0 (2.000) upoznajte ovaj fenomenalan program, koji vam omogućava više nego što ste očekivali

2. SUPERBASE (1.200)

Za vas malí i iznenađuje vas! KOMPJUTER BIBLIOTEKA FILIPA FILIPOVIĆA 41,32000 ČAČAK,tel. 032-31-20

KOMODOR 128 / KOMODOR 128
izašao je iz štampe:
"PRIRUČNIK ZA KOMODOR 128"

Na jednom mestu je detaljno objašnjen rad u sva tri moda: C128, C64 i CP/M. Kvalitetna štampa, plastificirani povez.
------- CENA 2.500 din -------
Takođe se možete pretplatiti na knjige koje su u pripremi:

1. "PROGRAMER'S GUIDE" (3.000)
2. "INTERN 128" (3.000)
3. "CP/M PLUS UPUTSTVO" (2.500)
4. "UPUTSTVO ZA 1571" (2.000)

KUPCI PRIRUČNIKA UŽIVAJU POPUST NA BUDUĆA IZDANJA OD 20%

"KOMPJUTER BIBLIOTEKA" FILIPA FILIPOVIĆA 41, 32000 ČAČAK telefon 032 - 31 - 20

SPEKTRUM Profesionalni prevodi za programere u mašinskom kodu MAŠINAC ZA POČETNIKE (1.200) DISASEMBLIRANI ROM (1.400) NAPREDNI MAŠINAC (1 400) U kompletu (3 600) Uputstva za uslužne programe DEVPAC MEGA BASIC BETA BASIC ARTIST MELBOURNE DRAV MONITOR DISASEMBLER EDITOR ASEMBLER FIFT DUILL TASWORD I LEONARDO Pojedinačno (500) U kompletu (34 000) Uputstvo za ELITE (900) Isporuka za 24 časa KOMPJUTER BIBLIOTEKA FILIPA FILIPOVIĆA 41 32000 ČAČAK, telefon 032 31 20

RAZNO

o Projektujem sve vrste elektronskih uređaja hardver i softver za mikroprocesore MC6800 Z80 I MC 68000 Zoran Kovačević dipl inž Ž. Popovića 45 15000 Šabac tel 27-617

o Zaštitite svoje računalo od praha prekrivači od kvalitetnog skaja po ugodnim cenama C128 — 2000 C-64 amstrad — 1500 din za ostali hardver navesti mjere isporuka brza. Peter Rotovnik, Aškerčeva 8 63325 Šoštanj tel 063/861 146

o Prodajem printer border VC 1520 časopise Vunx Chan E10 MT i Electronic po ceni od 450 din po komadu Meštrov Dragan Rapska 26—I 41000 Zagreb

o Prodajem literaturu za komodor 128 komodor 64 Sve o modemima robotici umjetnoj inteligenciji Ujedno prodajem disk drajv VIC 1541 te uređaj za zvuk Vo ce Master za C-64 Nikočević Miodrag Matije Gupca 64 41210 Zabok, tel 049/21-272

o ŠARP PC 1500 programi literatura o mašinskom jeziku (skripta 1 2 i 3) i hardver Eksterna memorija do 28 KB za bejzik Kesler Viktor Rumenačka 106-° 21000 Novi Sad tel 021/334-717

o DŽOJSTICI
Vrlo kvalitetna džojstika sa originalnim dlekom 4+4 pravca okidanje sa vrha palice za komodor amstrad atari i spektrum možete dobiti za svega 3500 din Za spektrumovce posebna ponuda. Kako da priključite džojstik bez interfejsa i uštedite do milion starih dinara?! Spektrumovci slušajte i vi zvuk svojih najomiljenijih igara preko televizora (kao na komodoru) Komplet već pripremljenih delova sa detaljnim uputstvom za ugradnju u spektrum — 1400 din sa ugradnjom — 2000 din ERF modulator 5 i 12 V 2000 din Stojković Dušan i Goran Trgovski trg 2 37000 Kruševac, tel 037/29-550

o Prodajem ZX81 1K sa pripadajućom opremom 50 programa Quicksave (20 puta ubrzava snimanje i tačnost) Prodajem nov elektronski voltmetar (1000 d) kupujem elektronske komponente Denoš Čabe Ž. Zrenjanina 20 23272 Novi Bečej

o Profesionalni mrežni filteri sa otklanjanje svih električnih smetnji i zaštitu kompjutera. Profesionalni kompjuterski rečnici srpskohrvatski-engleski-nemački i obrnuto preko 3000 reči! Tel 021/27-530 od 20 do 22 časa

o Šeme i literaturu za sve kompjutere predajem Katalog besplatan Zvonimir Vistrička, Svačićev trg 2 41000 Zagreb, tel 041/440-521

Izrađujem profesionalne kutije montaža pločice bez vijeka Evropa format Lim galvansko zaštićen i obojen plastikom Više dimenzija prema katalogu. Izrađujem i druge profesionalne uređaje termometri signalizatori dvopoložajni regulatori laboratorijski ispravljači. Ispravljači za stanice Za sve aparate dajem garanciju Informacije i besplatan katalog Simonič Peter Dobro 41 63270 Laško

o Servisiram računare KOMODOR 64 I ZX SPEKTRUM Izrađujem EPROM programatore za ista računare. Programiram sve vrste EPROM-a. Barbarić Dinko, Dječjo odred 21 55000 Slavonski Brod tel 055/236-702

računari su uvek aktuelni

da li ste sigurni da vam ono o čemu smo pisali juče neće biti potrebno već sutra?



Računari 3

Računari u izlogu: AMSTRAD CPC 464 ELEKTRON, SPEKTRUM PLUS
Periferijska oprema. DISK JEDINICE
Programi koje treba imati NEKI BOLJI BEJZICI
Majstorije na računaru Spektrum — HALO DA LI JE TO MAŠINAC?
BBC/Elektron — ŠTA EJKORN NIJE REKAO?
Računar „galaksija" ROM 2 (umetak na 32 strana)
Katalog najboljih igara svih vremena za spektrum. PEDESET VELIČANSTVENIH
Generisanje naših slova u printerima „epson" ĆIRILICA NA „EPSONU"
Škola avanturističkih igara. UKLETI DVORAC I DRUGE BAJKE
Simulacija letenja. LETAČI BEZ DIPLOME
Računari u domaćoj radinosti PROGRAMATOR EPROM-a (2)

Računari 4

Računari i njihove zamke: IMATI ILI UMRETI
Naš test: ATARI 800 XL
Uporedni telet: „AMSTRAD" PROTIV „KOMODORA
Računari u akciji DISKETNA VEZA
Računari u poslovnoj primeni MAŠINA SA BEZBROJ LICA
Računari u školi: HILJADU ZATO ZA RAČUNARE
Umetnost programiranja MODERNE PROGRAMERSKE TEHNIKE
Majstorije na računaru. Spektrum — NOVE NAREDBE I FUNKCIJSKI TASTERI
Računari na tržišnom ispitu SPRINTERI U KUĆICI PUŽA
Nova klasa mikroprocesora DŽINOVI NA GLAVI ČIODE
Katalog najboljih igara za „komodor" PEDESET NAJBOLJIH
Programiranje na bejziku NOVE AZBUKE NA „KOMODORU" PROMENLJIVE BEZ TAJNI
Škola sistemskog softvera. PUT U SREDIŠTE ROM-a
Škola simulacije letenja. DOBAR LET ELEKTRONSKA PTICO
Računari u domaćoj radinosti GENERATOR TONA

Računari 5

Računari u izlogu DVOBOJ DŽINOVA
Periferijska oprema. NEKI NOVI — RADOST ČITANJA
Periferijska oprema. NEKI NOVI „EPSONI
Računari i obrada teksta. PISANJE BEZ MUKE
Sedam načina komuniciranja sa kompjuterom. TAKO ĆE GOVORITI RAČUNARI
Računari u akciji STROGO KONTROLISANI DISKOVI
Škola sistemskog programiranja. RASPODELA MEMORIJE
Umetnost programiranja. Spektrum — INTERAPTI U BEJZIKU
Majstorije na računaru. Nove naredbe na spektrumu — NA SVOJU SLIKU I PRILIKU
Programiranje u bejziku SPRAJTOVI NA „KOMODORU
U svetu slučajnih brojeva. ZEC IZ ŠEŠIRA
Računari u domaćoj radinosti PRIČA O FINOJ GRAFICI

Računari 6

Računari u izlogu ENTERPRAJZ — PREDUZIMLJIVOST SA ZADRŠKOM PROMAŠAJ NA JAPANSKI NAČIN. RAČUNARI IZLAZEĆEG SUNCA
Periferijska oprema MONITORI
Programeri na ispitu HAKERSKI ZAKONIK
Programi koje treba imati SLIKARI BEZ KIČICE
Računari i obrada teksta: PERO OD OSAM BITA
Majstorije na računaru komodor 64 — NOVE NAREDBE NA NOVI NAČIN
Umetnost programiranja, spektrum — NOVA KRUNA ZA STAROG KRALJA
Put u središte ROM-a. VELIKA VIDEO PREDSTAVA
Majstorije na računaru spektrum — EKRAN POD PRESOM
Programiranje u bejziku HAMLET U RAČUNARU — Kako kompjuteri donose odluke
Umetnost programiranja. SVIRKA NA „KOMODORU
Računari i igre BESMRTNOST I KAKO JE STEĆI
Računari u domaćoj radinosti spektrum — ROM OD SEDAM MILJA „galaksija — FINI HARDVER ZA FINU GRAFIKU

Računari 7

Naš test: „KANON" PROTIV „EPSONA
Računari u izlogu. komodor 128 — O „KOMODORU" SVE NAJLEPŠE
Periferijska oprema: SVETLOSNA PERA
Umetnost programiranja. HAKER NA USIJANOM LIMENOM KROVU
Računari u akciji BAZE PODATAKA
Programi koje treba imati NE DAJ SE INES!
Računari u poslovnoj primeni. MALI RAČUNARI U VELIKOJ PRIVREDI
SVE „SPEKTRUMOVE RUTINE (umetak na 32 strane)
Test sa sedrikom: PREMA SVECU I DRAJV
Majstorije na računaru. komunikacija sa periferijskim uređajima — PUTOVANJE U PROVINCIJE
Programiranje u bejziku FUNKCIJE POTPROGRAMI PROCEDURE
Grafika na računaru KAKO TO RADI „AMSTRAD
Programiranje na mašincu: spektrum — KODOVI IZ RUKAVA
Put u središte ROM-a. LIČNE STVARI BEJZIKA
Računari u domaćoj radinosti ROM OD SEDAM MILJA (2) BRISAČ EPROM-a

Računari 8

Naš test: GALAKSIJA PLUS
Računari u izlogu ATARI 520 ST
Periferijska oprema. modemi — SVET NA DLANU
Istorija računara. RAĐANJE PROGRAMSKIH JEZIKA
Akcije: EKRANSKI EDITOR I DRUGE BAJKE
SVE „KOMODOROVE" RUTINE (umetak na 32 strane)
Računari i matematika. spektrum — GRAFIČKO PREDSTAVLJANJE FUNKCIJA
Programiranje u bejziku animacija ekrana (amstrad) — NA VRH BRDA VRBA MRDA

Majstorije na računaru spektrum — MAŠINSKA VEZA
Škola logičkih igara (1): VOLITE LI PASKAL?
Računari u poslovnoj primeni obrada ličnih dohodaka — PLAVI KOVERTI IZ RAČUNARA
Računari i umetnost: STRIPOTEKA NA KOMPJUTERU

Računari 9

Računari u izlogu ŽIVELA AMIGA
Hakerski manifest. PRAVI PROGRAMERI NE GOVORE PASKAL
Periferijska oprema PALICE ZA IGRU LASERSKA VEZA
Programski jezici KOMA ZBOG KOMALA
Tehnike programiranja. BACITE PROZORE KROZ PROZOR
Operativni sistemi PRVI GEM ZA TRIPOE
Put u središte ROM-a. U SVETU EDITORA
Akcija: PRIČA O EDITORU
Makazama po ROM-u spektrum — GENS IZ ROM-a
Majstorije na računaru: KVADRATNI KOREN (I)
Majstorije na računaru spektrum — MAŠINSKA VEZA (2) — pozivanje mašinskih programa iz bejzika
Majstorije na računaru komodor 64 — AUTOMATSKI START
Računari i nauka. STIMULANTI I MODELI
Škola logičkih igara (2) POSLEDNJI UVEK DOBIJA
Računari i matematika. NE DIRAJ MOJE KRUGOVE
Numerički metodi: INTERPOLACIJA — NUMERIČKI KRIVULJAR
Računari u poslovnoj primeni Obrada podataka u RO NA RAČUNARU
Računari u domaćoj radinosti DOPING ZA Z80

Računari 10

Računari u izlogu AMSTRAD PCW 8256
Pisanje na računaru. MUKE SA TEHNOLOGIJOM
Računari u izlogu SPEKTRUM 128
Naš test: Voice master 2 — SVOGA GLASA GOSPODAR
Računari i obrazovanje: NA BALKANU NIŠTA NOVO
Periferijska oprema. SA DISKOM ILI NA NJEMU
Razgovor sa računarom: MIŠEVI I SVEĆI

Organizacija računara i potprogrami iz ROM-a. SVE „AMSTRADOVE" RUTINE
Majstorije na računaru komodor 64 — PROGRAMI IZ ROM-a
Majstorije na računaru komodor 64 — TURBO KOJI MOŽE SVE
Matematički softver EKSPONENCIJALNA FUNKCIJA
IGRE KOJE STE NAJVIŠE VOLELI
Softverska igraa. DAN PO JUTRU A JUTRO TMURNO
Put u središte ROM-a. JEZIČKI PROCESORI

Računari 11

Periferijska oprema HARD DISKOVI
Kompjuterske mozgalice NEUPISIVO UPISIVANJE
Naš test: VIDEO DIGITAJZER
Računari u izlogu. Superkompjuteri 86 — DŽINOVI U BOCI
Operativni sistemi „AMSTRADOV" KERNAL
Naš test: PODMLAĐENI BBC
Umetnost programiranja: NAJKRAĆI PUT
Majstorije na računaru: komodor 64 — SRPSKOHRVATSKI BEJZIK
Interfejsi komodor 64 — „KOMODOR" U MREŽI
Programiranje u bejziku: MATRICE I NIZOVI
Majstorije na računaru. spektrum — NOVI ROM
Numerički metodi. LAGRANŽOVI POLINOMI
Radionica logičkih igara. KAMENČIĆI NA KOMPJUTERU
Matematički softver HIPERBOLIČKE FUNKCIJE
Računari u poslovnoj primeni ZALIHE NA DISKETI
Udruženi programeri PROFI ADRESAR

Računari 12

Pogled izbliza. LICEM U LICE SA „ATARIJEM
Softverska igraa. NOVINAR U SVOJOJ KUĆI
Istorija računara IBM I SEDAM PATULJAKA
Umetnost programiranja: HOLANDSKA ZAŠTAVA
Operativni sistemi CP/M
Programiranje na mašincu. SAŽIMANJE TEKSTA
Kako se koristi DEVPAC MONS
Umesto umetka. „AMSTRADOV" KALKULATOR
Majstorije na računaru spektrum — OPERACIJA FILL
Majstorije na računaru, amstrad — KRALJEVSTVO ZA REGISTAR
Akcija komodor 64 programi se koriguju — SAM SVOJ PROGRAMER
Matematički softver LOGARITAMSKA FUNKCIJA
Udruženi programeri JEZIČKI PROCESORI U 10 LEKCIJA
Programiranje u bejziku SVIRKA NA AMSTRADU
Put u središte ROM-a: PROCEDURE U AKCIJI
Hajde da se igramo POKCE KOJE ŽIVOT ZNAČE ELITNA GROZNICA
Spektrumovci u nevolji SMRTONOSNA DIJAGNOSTIKA

Računari 13

Računari u izlogu. NEKI STARI GOSPODARI
Računari na sajmu: MIKROCOMPUTER '86
Računari u izlogu SUPERPERSONALCI U VAŠOJ KUĆI
Operativni sistemi. MALI DIV VELIKOG SRCA
Računari na testu Megastan — DŽIN IZ PLASTIČNE KUTIJE
Udruženi programeri ČASOPISI ZA HARDVERAŠE
Umesto umetka. DEVPAC GENS
Obrada ideja. MISLI KAO ŠTO PIŠEŠ
Akcije komodor 64 — bejzik iz eprom-a
Matematički softver AREA FUNKCIJE
Umetnost programiranja. TRIVIJALAN NAJTRIVIJALNIJI SKORO REŠIV ZADATAK
Tehnike programiranja. rekurzije — KAKO SE IZBAVITI IZ LAVIRINTA
Biblioteka programa. EKRANSKI EDITOR (1)
Radionica logičkih igara. KAKO POTKRESATI STABLO

računari 1 i 2 rasprodati

NARUDŽBENICA

Galaksija Bulevar vojvode Mišića 17 11000 Beograd

Molim vas da mi pouzećem pošaljete sledeće stare brojeve RAČUNARA 3, 4, 5 6 (za 20 dinara po primerku)
7 8 9, 10 11 (za 250 din po primerku)
12 (po ceni od 300 din) — zaokružiti odgovarajuće brojeve

Ime i prezime ____________________

Ulica i broj ____________________

Broj pošte i mesto ____________________

(Datum) (Potpis)

NAPOMENA Ukoliko ne želite da isecanjem narudžbenice oštetite svoj primerak „Računara" molimo da potrebne podatke ispišete na dopisnici ili u pismu i pošaljete na navedenu adresu

ekranski editor

Naredbe za podešavanja

FREEZE ili FR

Zamrzava dva gornja reda ekrana. To praktično znači da ih listanje programa i slične rednje neće obrisati

Dva gornja zamrznuta reda su izuzetno praktična da u njih upišete, recimo, neku matematičku formulu koju često koristite u programu — kasnije, kad god vam zatreba, možete da je iskopirate. Ta dva reda možete, takođe, da iskoristite kao neku vrstu elektronskog bloka za čuvanje poruka poput „Pazi na liniju 200" ili „DF SZ = 23659"

UNFREEZE ili UNFR

Odmrzava dva gornja reda ekrana.

COLOR n,m ili CL n,m

Parametri n — boja pozadina (paper i border)
m — boja slova (ink)

Postavlja boju pozadine i slova

REPEAT n,m ili REP n,m

Parametri n — brzina za sve tastere osim CAPS/5 i CAPS/8
m — brzina samo za CAPS/5 i CAPS/8

Podešava brzinu automatskog ponavljanja ako je taster stalno pritisnut

Kada se učita editor n ima vrednost 4 a m=2. Ako vam je to presporo, probajte REP 2,1. S druge strane, ako volite da čekate, otkucajte REP 6,4 ili REP 8,5

Zašto REPEAT naredba ima dva parametra? Jednostavno zato što ekran ima 51 poziciju horizontalno, a samo 20 vertikalno. U takvim uslovima bilo bi vrlo neprirodno da se kursor pomera istom brzinom i vertikalno i horizontalno (otkucajte REP 1,1 pa se i sami uverite)

FTIME [n] ili FT [n]

Parametri n — vreme u sekundama, ako se na navede vrednost = 60

Definiše vreme posle kojeg kursori počinju da fleširaju, ako se ne pritisne nijedan taster

Kada se učita editor FLASH TIME je postavljen na 60 — ako jedan minut ne pritisnete nijedan taster, kursori počinju da fleširaju. To je zato da ne biste morali pogledom da tražite kursore po ekranu, jer se pretpostavlja da ceo taj minut niste gledali u ekran (da ste gledali u ekran, sigurno biste i pritisli neki taster)

BEEP ili BP

Svaki pritisak na neki taster biva propraćen sa jednim „blip"

UNBEEP ili UNBP

Poništava BEEP stanje (videti BEEP)

KB+

Predefiniše tastaturu da odgovara „spektrumu plus". Da bi se označilo to stanje, u status liniji, sasvim desno, pojavića se jedno malo +. (Više detalja, na kraju serije, u tekstu „Organizacija tastature".)

KB-

Vraća tastaturu da odgovara običnom spektrumu (videti KB+)

ZOP

ZERO OPTION. Predefiniše karakter set tako da cifra nula i slovo „O" izgledaju malo drugačije. Autoru se čini da je tako poboljšana čitljivost, naročito linijskih brojeva

ZOFF

Poništava ZERO OPTION

Ako ste iz radoznalosti otkucali ZOP pa vam se rezultat nije svideo, onda je ovo prava naredba za vas

Naredbe za pamćenje slike

STORE ili STO

STORE SCREEN. Naredba STORE nalaže editoru da zapamti sliku koja se trenutno nalazi na ekranu. Od tog trenutka naredba HELP više neće pokazivati „HELP SCREEN" (videti HELP) već tu vašu sliku, a moguće je koristiti i RECALL naredbu

Za šta ovo može da koristi? U mnogim situacijama vrlo je zgodno stalno imati pred očima listing nekog važnog programskog segmenta ili potprograma. Tada je kombinacija STORE-/HELP zlata vredna

RECALL ili RCL

RECALL SCREEN. Vraća na ekran sliku koja je bila sačuvana pomoću STORE naredbe. Naravno, time slika neće biti izbrisana iz memorije

Napomena: RECALL daje grešku „INCORRECT STATEMENT" ako nikakva slika nije bila upisana u memoriju pomoću STORE naredbe

EXCHANGE ili EXC

EXCHANGE SCREENS. Slika sa ekrana ide u memoriju, a slika iz memorije se vraća na ekran

Napomena: EXCHANGE DAJE GREŠKU „INCORRECT STATEMENT" ako nikakva slika nije bila upisana u memoriju pomoću STORE naredbe

Naredbe za rad sa mikrodrajvom

MDF n

Parametar n — broj mikrodrajva, ako se ne navede vrednost = 1

Za razliku od svih drugih naredbi, MDF deluje u bejziku a ne u editoru. Dejstvuje tako što automatski ispisuje zvezdice, navodnika, tačka zareze i sve ostalo što ide iza SAVE, LOAD, VERIFY, FORMAT i EARSE naredbe kada se radi sa mikrodrajvom

Otkucati MDF 0 je isto što otkucati MDFOFF — isključuje se MDF stanje

MDFOFF

MICRODRIVE FACILITY OFF. Poništava MDF stanje (Videti naredbu MDF)

Vladimir Kostić

Numerički metodi za mikroračunare

Autori Nenad Mladenović Veljko Spasić Milorad Jovanović, recenzant dr Lav Ivanović docent PMF Izdavač NIRO „Tehnička knjiga I Zavod za izdavanje udžbenika, Beograd 1986 Strana 185 cena 1 850 dinara tiraž 4 000 primeraka

Autori mr Nenad Mladenović matematičar koji radi kao asistent na Fakultetu organizacionih nauka u Beogradu mr Veljko Spasić, matematičar koji radi kao istraživač u Centru za multidisciplinarne studije u Beogradu i Milorad Jovanović matematičar koji radi kao profesor programiranja u Beogradu, izložili su u knjizi programe kojima se na mikroračunarima komodor 64 i „spektrum realizuju metode numeričke analize N Mladenović je jedan od koautora knjige „Kućni kompjuteri (algoritmi i programi) a V Spasić — BASIC za mikroračunare obe u izdanju Tehnička knjiga dok je M Jovanoviću ovo autorski prvenac Reauma se, sam izbor računara na kojima su programi realizovani kazuje da ambicija autora i izdavača nije bila da objave naučno delo već da približe metode numeričke matematike širem krugu korisnika koji se susreću sa ovim problemima bilo u okviru školskih predmeta bilo što ih koriste kao sredstvo za obavljanje složenijih projekta i istraživanja Stoga i ne dajemo naučno-stručni prikaz knjige već iznosimo samo informacije koje bi mogle biti interesantne za potencijalne čitaoce ovako koncipirane knjige

Na početku je u Bekusovoj notaciji dat pregled naredbi komandi i funkcija bejzik jezika za „komodor i „spektrum a zatim su prikazane metode interpolacija rešavanja sistema linearnih algebarskih jednačina računanje karakterističnih vrednosti i rešavanje nelinearnih jednačina Svaka od metoda izložena je u najkraćim crtama objašnjen je algoritam korišćen za njenu implementaciju na računaru i priloženi su test primer i listing programa za komodor ili spektrum Ova shema je dosledno korišćena za sve metode ali se ipak može zapaziti da je knjigu radilo više autora koji nisu do detalja uskladili stil izlaganja Kako su korišćene samo osnovne mogućnosti bejzika programi se jednostavno mogu modifikovati za korišćenje na drugim računarima, a profesionalni način kreiranja programa dozvoljava da se sa lakoćom mogu vršiti i druge vrste izmena Sve su ovo pogodnosti koje omogućavaju da se izloženi programi mogu koristiti ne samo u nastavi numeričke analize već i u nastavi programiranja

Ipak treba imati u vidu da knjiga mada nosi naziv Numeričke metode pokriva samo njihov deo i da u programima nije dobro izabran kriterijum završetka izračunavanja Grafička rešenja i kvalitet štampe su na zavidnom nivou.

Commodore za sva vremena

Autori Dragan Tanaskoski Stevan Milinković, Vladimir Janković, recenzent Ladislav Rupnik, izdavač Samostalno izdanje autora, Beograd 1986 Strana 336 cena 3 600 dinara, tiraž 6 000 primeraka

Autori Dragan Tanaskoski mašinski inženjer koji radi u institutu za fiziku i Stevan Milinković i Vladimir Janković inženjeri elektronike koji se računarima profesionalno bave u Institutu „Boris Kidrič' u Vinči pripremili su za vlasnika komodora knjigu koja im može pomoći da ovladaju većinom njegovih mogućnosti D Tanaskoskom i V Jankoviću ovo nije prvo samostalno izdanje Oni su kostori i izdavači uspešno prodavane knjige Spektrum priručnik a Dragan Tanaskoski je našim čitaocima poznat i kao koautor programa Eastenglish nagrađenog na Galaksijinom konkursu za najbolje YU programe

Počevši od osnovnih pojmova kao što su bit i bajt i načina uključivanja računara i korišćenje ekranskog editora autori izlažu korisnicima komodora sve one informacije koje bi se inače morale tražiti u mnoštvu za naše prilike vrlo skupih knjiga na stranim jezicima Većina bitnih informacija iz nezaobilaznog Programmer s Reference Guidea (po ceni od 20$) može se korektno interpretirane naći u ovoj knjizi Zoran Životić i Nevenka Spasić, koji su knjigu čitali prepoznali su mnoge izvore koje su autori koristili ovu knjigu treba pre svega prihvati kao priručnik koji na jednom mestu iznosi informacije bitne za korišćenje popularnog komodora a ne kao iznošenje novih saznanja — utoliko pre što su svi izvori uključujući i beleške sa predavanja profesora Dujmovića na Elektrotehničkom fakultetu navedeni u literaturi istini za volju mora se reći da stvari koje su ostale nedorečene u Reference Guideu i drugim izvorima nisu ni ovde pojašnjene

Posebno poglavlje posvećeno je mogućnostima i korišćenju sajmona bejzika a programiranju na mašinskom jeziku kao i svim ostalim celinama prethodi mali teoretski uvod koji objašnjava pojmove koji će se koristiti Za ambicioznije čitaoce verovatno će posebno biti zanimljiva poglavlja posvećena organizaciji memorije i upotrebe ROM rutina u hardveru Prezentirane su i konstrukcija centronika interfejsa RS 232 interfejsa modema, eprom programatora i ROM modula

Mada se načinu izlaganja materija mogu staviti i neke zamerke ona nisu suštinske i čitalac se bez obzira na njih u otkrivanju tajni komodora može osloniti na ovu knjigu Očigledno je da su se velikog posla pisanja sveobuhvatnog priručnika po meri jednog inženjera koji se prethodno nije bavio računarima prihvatili znaci koji dobro poznaju materiju koju izlažu

Sistemski softver

„Amstrad"

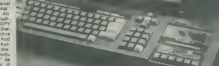
ni na nebu, ni na zemlji

Operativni sistem „amstrada" ne prestaje da iznenađuje neumorne istraživače. Otkrivajući tajne „amstradovog" kalkulatora, ukazao se još jedan čudan stanovnik donjeg ROM-a. Radoznalim čitaocima sigurno nije promakao jedan (nedokumentovan) ulaz (BO3A) koji se nalazi između glavnog i kalkulatorovog džemp-bloka. Oni koji su otišli dalje (u ROM), verovatno su bili neprijatno iznenađeni izuzetno razuđenim algoritmom. I, zaista, radi se o jednom od najkompleksnijih potprograma unutar operativnog sistema. Reč je o editoru.

Amstradov editor je u osnovi linijski mada je ostavljena mogućnost šetanja po čitavom ekranu ili preciznije po trenutno važećem prozoru To znači da se tekst može editovati na bilo kojoj poziciji (tekst" označava string dužine 255 znakova) Ali ono po čemu se „amstradov" editor razlikuje od običnih linijskih editora je COPY opcija Ona omogućava na vrlo specifičan način (koji je dobro poznat vlasnicima BBC-ja) izlazak iz editovane linija Kod običnih linijskih editora postoji jedan kursor, dok u našem slučaju postoje dva Jedan služi za pisanje (označava poziciju na koju se smeštaju slova „očitana" sa tastature) a drugi za čitanje ' Drugi ili COPY kursor može se slobodno pomerati po čitavom prozoru Pritiskom na specijalni taster COPY čita se slovo koje se nalazi ispod COPY kursora i upisuje na mesto kursora za pisanje Dakle postoje dva izvora teksta — tastatura i prozor u kom se vrši editovanje (na žalost, ovaj princip nije poštovan do kraja tako da ne postoji mogućnost da disk ili kasetofon budu izvori) Ovakav editor je sasvim dovoljan za rad sa programskim jezicima, pri čemu je posebna pogodnost mogućnost editovanja direktnih komandi Priložena je i tabela komandi koje editor razume (reč tekst označava liniju koja se edituje a reč linija jednu fizičku liniju na ekranu)

Pogled na datu tabelu ne otkriva naročito bogatstvo naredbi mada one omogućuju prilično komforan rad Na žalost amstradov editor je programski skoro potpuno zatvoren To znači da mu se akcija mogu izmeniti samo na nekim neprirodnim mestima — u ovom slučaju to su linkovi za grafičko štampanja karaktera i uključivanje-isključivanje kursora Ako neko bude želeo da prilagođava ovaj editor svojim potrebama ili dodaje nove naredbe biće mnogo lakše da napiše novi

Ugrađeni bejzik interpreter koristi prikazani editor za unošenje i ispravljanje programa Na mikroračunarima je ustaljena praksa da se editori implementiraju u okviru programskih jezika, pa je prilično čudno što se ovaj nalazi u okviru operativnog sistema Naravno ovo je pogodna okolnost jer editor može da se koristi i za neka druge programska jezike (šteta je što to HISOFT nije uradio za paskal i asembler — ili nisu imali potrebnu dokumentaciju za ovaj već ugrađeni, ili neće da odustanu od svojih



OD CR (ENTER)	kraj editovanja i indikacija za prihvatanje izmena tj teksta
10 DLE (CLR)	briše znak ispod kursora a tekst s desne strane se pomera ulevo
7F DEL (DELETE)	briše prvi znak s leve strane a tekst s desne strane se zajedno sa kursorom pomera ulevo
EO COPY (COPY)	znak ispod kursora za čitanje se prepisuje u editovanu liniju
E1 INS (CTRL TAB)	flip flop koji određuje insert ili overwrite mod
FO WUP (↑)	kursor za pisanje nagore
F1 WDN (↓)	kursor za pisanje nadole
F2 WLT (←)	kursor za pisanje ulevo
F3 WRT (→)	kursor za pisanje udesno
F4 RUP (SHIFT ↑)	kursor za čitanje nagore
F5 RDN (SHIFT ↓)	kursor za čitanje nadole
F6 RLT (SHIFT ←)	kursor za čitanje ulevo
F7 RRT (SHIFT →)	kursor za čitanje udesno
F8 BEG (CTRL ↑)	kursor za pisanje na početak teksta
F9 END (CTRL ↓)	kursor za pisanje na kraj teksta
FA STA (CTRL ←)	kursor za pisanje na početak linije u kojoj se nalazi
FB FIN (CTRL →)	kursor za pisanje na kraj linije u kojoj se nalazi
FC BRK (ESCAPE)	kraj editovanja i indikacija da se korisnik odlučio da ulazni tekst ostane neizmenjen

Pri pokušaju da se izvede nešto nedozvoljeno (izlazak kursora za pisanje iz teksta koji se edituje ili recimo pokušaj brisanja kada je linija prazna) javlja se zvučni signal Ukoliko je editovani tekst prazan kursor za pisanje se može slobodno pomerati po čitavom prozoru

linijskih editora) A zgodno je imati i kalkulator pri ruci

Dakle, sve je to lepo i krasno ali pitamo se na šta je utrošeno punih 16K (do poslednjeg bajta) gornjeg ROM-a Ono što nudi interpreter konstruktori „galaksija" bi verovatno smestili u neka 4 kilobajta Po svemu sudeći programeri Lokomotiv softvera se nisu baš pretrgli pišući bejzik za „amstrad" jer kako inače objasniti tolike razliku u kvalitetu ova dva programa, ugrađena u isti računar Ostaje nam samo da čekamo dan kada će se proizvođači računara poštenije odnositi prema svojim kupcima

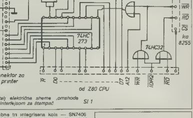
Dejan Muhamedagić

paralelni interfejs za štampač

Rešenje paralelnog interfejsa koji je primenjen u računarima amstrad ima jeden neprijatan nedostatak koji u nekim slučajevima može u mnogome da ograniči rad printeru se šalju samo sedem bite dok je izvod na konektoru za osmi bit spojen na masu (logička nula) Kako su svi standardni alfanumerički znaci predstavljeni ASCII kôdom koji je menji od 128 (znači mogu se definisati sa sedem bita) to je za štempanje normalnog teksta ovakva veza sasvim zadovoljavajuća Problem nastaje kad se žele štampati specijalni karakteri čiji je ASICII kód definisan sa osam bite e koji se pored standardnog sete znakova takođe jelaze u ROM-u printera ili se želi formireti sopstveni set znakova Nedostatak osmog bita kod hard-copy programe priilčno usporava red

Postoji više nečina za otklanjanje ovog nedostatke od kojih su neki tako jednostavni de je čudno da ih proizvodač nije ugradio Kako varovatno ima veći broj vlasnika ovih računers kojima smeta, nedostatak osam bita ne izlazi ze printer, ovde se prikazuju dva jačina kako se on može dovesti do printera

Da bi izmena koje se navode bile jasnije, treba najpre videti kako je ovaj interfejs rešen u fabričkoj izradi računare Made je u računaru ugradeno integralno kolo 8255 (paralelni ulazno-izlazni interfeje) ono je namenjeno za komunikaciju se tastaturom kasetofonom i generatorom zvuka Samo se jeden bit (šesti) kanala B koristi za Centronics „busy" signal Osnovu paralelnog interfejsa kod ovih računara sačinjava 8-bitni letch 74HC273 čijih je sedam bita iskorišćeno za podatak (kôd karaktera) a osmi služi ze „strobe" signal (poruka printeru da ga na ulazu čeka podatak) Ne slici 1 je prikazan deo električne sheme računara koji se odnosi ne interfeja Propuštanje podatka na izlaz ovog leča vrši se aktiviranjem edrese &EF×× pri čemu se podatak sa magistrale podataka šalje na izlaz za printer Što se tiče programske rutine u ROM-u računara ona je data na slici 2 a poziva se iz bejzika sa PRINT #8



Detalj električna sheme ,omshode sa interfejsom za štampač *Sl 1*

Novi interfejs . .

Kako se iz programe vidi omogućeno je da se ista naredba konsti i za startovenje programa koje korisnik sam piše jer je veza ostvarana preko pozivne rutine operativnog sistema &BD2D Iz istog listinga se takode vidi (linije 210 i 260) da je ovim programom eliminisan osmi bit što govori o tome da se ovaj deo programa ne može koristiti u našem slučaju bez izmena celog ROM-a, što je, sigurno potpuno neprihvatljivo

Jeden od načina da se na izlaz za printer izvedu svih osam bita je da se napravi poseban interfejs koji bi se prikijučio na port za proširanje (za one koji ne žele da vrše nikakva ispravka u samom računaru), se u samom računaru ostvaruje povezivanje i smeštaj elemeneta Pri tom su potrabne tri integrisana kola — SN7406 SN74LS32 i 8255 — i tri elektrolita od 10 μ F uz odgovarajuće konektore Sheme ovog dodetka je date na slici 3 Kao što se vidi napajanje (+5 V) se uzima iz računara, tako da nije potreban dodatni stabilizovani ispravljač Ovim rešenjem se dobija interfeja sa 24 ulazno-izlaznih kanala kod kojih je samo 9 koriste za vezu sa printerom dok se drugi mogu proizvoljno koristiti za druge namene Same činjenica da na raspolaganju ostaje 15 slobodnih kanala za razne upravljenje ili komunikaciju sa drugim uradejima opravdava investiciju i napor da se ovakav interfeja napravi

Većinu gornjih adresa adresne magistrale (A8 A15) koristi sam računar Korisniku za proširenja ne raspolaganju ostaju samo A8 A9 i A10 odnosno adrese &FB &FA i &FB U konkretnom slučaju je najbolje koristiti adresu &FB jer se aktiviranje vrši samo logičkom nulom ne A10 Kako je za komunikaciju se izlazom potrebno više od jedne adrese mora se u spoljašnjem interfejsu dekodirati i sadržaj ragistre C zajedno se ragistrom B odnosno neophodne je uvak 16-bitna port adrese Iz raspoloživih adresa su pored A10 iskorišćene A0 i A1 (slika 3) tako da će biti

&FB00 — Kanal A ulaz/izlaz
&FB01 — Kanal B ulaz/izlaz
&FB02 — Kanal C ulaz/izlaz
&FB03 — upravljanje pomoću upravljačkog bajta '

Upravljanje (odrađivanja stanje pojedi-

Bit 7 1 — oznaka da se podatak koristi kao upravljački bajt
Bit 6: Izbor medusa grupe A 00-modus 0; 01-modus 1
Bit 5. 10 ili 11-modus 2
Bit 4 Kanal A 1 ulaz 0-izlaz
Bit 3 Kanal C2 (gornje vrednosti) 1-ulaz; 0-izlaz
Bit 2 Izbor medusa grupe B. 0-modus 0. 1-modus 1
Bit 1 Kanal B 1 ulaz; 0-izlaz
Bit 0. Kanal C1 (donje vrednosti) 1-ulaz, 0-izlaz

```
Hisoft GENA3 | Assembler  Page  1

Pass 1 errors: 00

07E6              10          ORG  #07E6
07E6  21EC07      20 RESETP:  LD   HL UL1
07E9  C3A00A      30          JP   UL3
07EC  03          40 UL1:     DEFB 3
07ED  F108        50          DEFW 08F1
07EF  C3F007      60          JP   WAITPR
07F2  C5          70 PRINTC:  PUSH BC
07F3  CDF007      80          CALL WAITPR
07F6  C1          90          POP  BC
07F7  C9         100          RET
07F8  013200     110 WAITPR:  LD   BC #32
07FB  CD1D08     120 UL2:     CALL BUSYPR
07FE  3007       130          JR   NC SENDPR
0800  10F9       140          DJNZ UL2
0802  0D         150          DEC  C
0803  20F6       160          JR   NZ UL2
0805  B7         170          OR   A
0806  C9         180          RET
0807  C5         190 SENDPR:  PUSH BC
0808  06EF       200          LD   B #EF
080A  E67F       210          AND  #7F
080C  ED79       220          OUT  (C) A
080E  F680       230          OR   #80
0810  F3         240          DI
0811  ED79       250          OUT  (C) A
0813  E67F       260          AND  #7F
0815  FB         270          EI
0816  ED79       280          OUT  (C) A
0818  C1         290          POP  BC
0819  37         300          SCF
081A  C9         310          RET
081B  C5         320 BUSYPR:  PUSH BC
081C  4F         330          LD   C A
081D  06F5       340          LD   B,#F5
081F  ED78       350          IN   A (C)
0821  17         360          RLA
0822  17         370          RLA
0823  79         380          LD   A C
0824  C1         390          POP  BC
0825  C9         400          RET
                 410 ;
0A8A             420          ORG  #0A8A
0A8A  4E         430 UL3:     LD   C,(HL)
0A8B  0600       440          LD   B,0
0A8D  23         450          INC  HL
0A8E  5E         460          LD   E (HL)
0A8F  23         470          INC  HL
0A90  56         480          LD   D,(HL)
0A91  23         490          INC  HL
0A92  EDB0       500          LDIR
0A94  C9         510          RET

Pass 2 errors: 00                         Sl 2

Table used:  100 from  166
```

Ugradnja paralelnog (Centronics) interfejsa u kućne računare „amstrad/šnajder" CPC omogućava direktno priključivanje većine printera, bez potrebe za ne baš jeftinim dodacima, koje zahtevaju drugi računari slične klase. Ovo je, sigurno, jedna od prednosti računara „amstrad/šnajder", jer se ozbiljnija primena ne može ni zamisliti bez printera. Na žalost, konstruktori ovog popularnog računara doneli su jednu čudnu odluku koja zagorčava život njegovim korisnicima. Sve se, međutim, može popraviti . . .

nih kanala) integrisanog kola 8255 vrši se pomoću upravljačkog bajta i to po sledećoj šemi

U konkretnom slučaju upravljački bajt ima vrednost &83 što znači da je kanal A predviđen za izlaz kanal C1 za ulaz kanal C2 za izlaz kanal B za ulaz uz modus 0

Opis sema shema interfejsa je jednostavan signal za izbor čipa se dobija direktno se adrese A10 mikroprocesora Z80 odnosno porta za proširenja Takođe se direktno povezuju i adrese A0 i A1 Za razliku od upisivanja i čitanja memorije ovde su neophodna oba signala IOWR i IORD Oni se dobijaju preko dva OR kola iz WR RD i IORQ Sistemski RESET procesora je aktivno nisko tako da se mora invertovati za upravljanje kolom 8255 D0 D7 se porta se pajaju direktno sa ulazima za podatke 8255 Napajanje svakog integrisanog kola se mora blokirati letovanjem elektrolita na konektor za printer u računaru (ukoliko se interfejs ugrađuje u računar) ili na poseban konektor

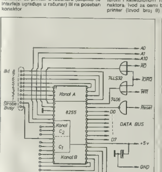
Pored ovog hardverskog dodatka neophodan je i odgovarajući program kojim će se njim upravljati Ovaj program je dat na sl 4

... ili mali rez

Drugi način je daleko jednostavniji On se preporučuje svim onima koji se ne plaše da vrše izvesne dorade i izmene na samoj štampanoj ploči računara Ove izmene su toliko male da ih mogu izvršiti i oni koji se nikad nisu bavili elektronikom Važno je samo biti malo pažljiv pri letovanju da se ne oštete drugi vodovi ili samo integrisano kolo Za ovu izmenu je od dodatnog materijala potrebno samo nekoliko centimetara duge izolovane žice i, naravno odgovarajuća električna lemilica (do 25 W)

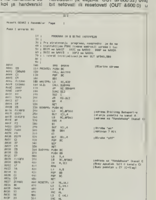
Suština izmene se sastoji u sledećem najpre se odviju 6 zavrtnjeva sa donje strane računara i odvoji gornji deo sa tastaturom i kasetofonom rastavljajući dva konektora Izvod za osmi bit na konektoru za printer (izvod broj 9) koji je hardverski spojen sa masom treba oštrim sečivom razdvojiti od mase (sl 5) pri samom dnu Zatim se izolovanom tankom žicom (najbolje licnastom) izvrši spajanje ovog izvoda (iznad prekida) sa vodom na štampanoj ploči koji ide od pina 12 integrisanog kola 8255 i to najbolje kod otvora gde se veza prenosi na drugu stranu Pri letovanju treba paziti da se ne oštete veze na štampanoj ploči, kao i na to da se kalaj ne razlije po izvodu konektora i — izmena je gotova

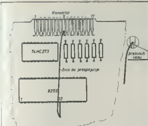
Posle ove hardverske izmene potreban je i odgovarajući program koji šalje osmi bit na izlaz kada je to potrebno Sistemska rutina u ROM-u računara tretira naime Centronics izlaz još uvek kao 7-bitni interfejs Opisanom izmenom je osmi bit konektora printera spojen sa petim bitom kanala C integrisanog kola 8255 koji je sistemski predviđen za slanje podataka na kasetofon Kako printer i kasetofon nikad ne rade istovremeno moguće je odgovarajućom naredbom (iz bejzika OUT &F600 32) ovaj bit setovati ili resetovati (OUT &F600 0) u



Električna shema novog interfejsa

Sl 4





Sl 5

Hardverska modifikacija amstrada

```
Hisoft GENA3 I Assembler  Page     1

Pass 1 errors: 00

                 10 ;      PROGRAM ZA 8-BITNI INTERFEJS 2
                 20 ;
                 30 ; Pre startovanja  programa, neophodno  je da
                 40 ; sa instrukcijom POKE izmeni sadrzaj adrese
                 50 ;  BDF1 sa  "C300A0"
                 60 ;
                 70 ;
0000             80          ORG  #A000          ;Startna adresa
0018             90 BUSYPR:  EQU  #0018
A000  013200    100 WAITPR:  LD   BC,#32
A003  CD1800    110 UL2:     CALL BUSYPR
0000  3007      120          JR   NC SENDPR
0000  10F7      130          DJNZ UL2
0000  0D        140          DEC  C
0000  20F6      150          JR   NZ,UL2
0000  B7        160          OR   A
0000  C9        170          RET
A000  C5        180 SENDPR:  PUSH BC
0000  06F6      190          LD   B,#F6          ; Kanal C IC 8255
A012  CB7F      200          BIT  7,A            ; Karakter veci od 127?
0000  2806      210          JR   Z,DALJE        ; ako nije- preskoci
A01A  F5        220          PUSH AF
A017  3E20      230          LD   A,#20          ; 5  bit setovan
A019  ED79      240          OUT  (C),A          ; slanje podatka na kanal C
A01B  F1        250          POP  AF
A01C  01EF00    260 DALJE:   LD   BC,#EF
A01F  E67F      270          AND  #7F
A021  ED79      280          OUT  (C),A          ; podatak bez STROBE na izlaz
A023  F680      290          OR   #80
A025  F3        300          DI
0000  ED79      310          OUT  (C),A          ;Strobe "on"
A000  E67F      320          AND  #7F
A02A  FB        330          EI
A02B  ED79      340          OUT  (C),A          ;Strobe "off"
A000  F5        350          PUSH AF
A000  06F6      360          LD   B,#F6          ; Kanal C IC 8255
A030  3E00      370          LD   A,0
A032  ED79      380          OUT  (C),A          ; brisanje 5. Bita kanala C
A034  F1        390          POP  AF
A035  C1        400          POP  BC
0000  37        410          SCF
A037  C9        420          RET

Pass 2 errors: 00
                                    Sl 6
Table used:    74  from   198
```

zavisno da li se šalje osmobitni ili sedmobitni podatak. Korišćenje bejzika je ne samo sporo već i prilično neudobno pa je zato urađen program u mašincu koji se učitava u RAM na adresu &A000, a prikazan je na slici 6. Pritom se mora izmeniti i adresa skoka na adresi &BDF2 i to sa poke &FD2,0 i poke 3BDF3 &AO

Svetislav Zahar

Kako se unosi mašinski program u računar? Odgovor je vrlo jednostavan na isti način kao što to radite sa bejzikom. Mnogi asembleri imaju svoj editor pa odmah na početku treba savladati načine kako da se izlistа grupa linija kako da se ubaci ili izbaci neke od njih itd. Bejzik editorom smo uglavnom svi zadovoljni na njega smo navikli pa nema potrebe da u tom delu tražimo nešto drugo. Dakle linije se pišu sa linijskim brojem program lista običnim LIST, snima se SAVE itd. Dovoljno je da napišete na primer

100 LDA #10

i dobili ste jednu liniju asembler programa. Da stvar bude lepša ispred i iza nje se može nalaziti i običan bejzik program što za sada može izgledati bez značaja ali može i te kako biti korisno u praktičnom radu. Kako da se obeleži deo koji predstavlja asembler da bi ga PA preveo u niz mašinskih instrukcija? I ovo je vrlo jednostavno ispred prve linije asembler programa ubacite SYS28672 (ako koristite kertridž verziju onda SYS32768). Najjednostavniji primer je dat na slici 1 P1.

Ako zadate RUN bejzik će obaviti svoj posao u liniji 100 a zatim (110) asembler preuzima kontrolu i sledeće dve linije prevodi u niz brojeva 169 0 141 33 208 96 koji predstavljaju mašinski program. Ako hoćete da vidite šta ovaj program radi ostaje da ga pozovete bejzik naredbom SYS. Sve je to jednostavno ali gde je program u memoriji? Odgovor će vas iznenaditi NIGDE. Zadatak dobrog asemblera nije samo prevođenje programa već i olakšavanje rada programeru. Jedna od takvih olakšica je i mogućnost da se prođe kroz program proveri da li je sve u redu ali da se smeštanja koda u memoriji ne obavi. Sada već nazirete da asembler može mnogo više ali mu se mora staviti do znanja šta od njega očekujemo. Zato su uvedene posebne naredbe takozvane pseudo instrukcije. Pišu se skraćeno sa tri slova a da bi se razlikovale od asembler instrukcija mikroprocesora ispred se stavlja tačka

Pseudonaredba OPT

Osnovne naredbe ovog tipa je OPTION koje se piše OPT. Njenim parametrima se asembleru naređuje šta da uradi pri prevođenju da li da samo proveri program da li da kod smesti u memoriju itd. Kod PA je ovaj deo najkomplikovaniji jer OPT može imati do tri parametra u proizvoljnim kombinacijama a neki od njih i više oblika. Zato ćemo za početak upoznati samo jedan — OO (dva slova oo). Ako se dakle negde u izvornom programu nađe OPT OO asembler će od tog trenutka započeti smeštanje mašinskog koda u memoriju. Primetite da smo upotrebili negde u programu. Pseudo naredbe se mogu naći na proizvoljnom mestu i pojavljivati proizvoljan broj puta u jednom programu. Profi asembler ne postavlja nikakva ograničenja pa je za sada dovoljno da zapamtite da je upotreba potpuno slobodna i da treba da bude prilagođena isključivo onome što vi od asemblera očekujete

Izvršna adresa

Poznavanje ove opcije OPT naredbe (jasno uz poznavanje samih mašinskih instrukcija) potpuno je dovoljno da napišete program koji radi. Njome smo obezbedili da se mašinski program nađe u memoriji. Gde će međutim on biti smešten? Ako se ne zada početna adresa, profi asembler će podrazumevati da treba da smesti kod od adrese 49152 ($C000). Ako želite da počne od neke druge adrese upotrebite naredbu koja jedina predstavlja izuzetak po načinu pisanja i izgleda ovako

* = početna adresa

Znak * je specijalna varijabla kojoj znakom = dodeljujete početnu vrednost. Kada asembler započne sa prevođenjem stalno će menjati njen sadržaj prema tome do koje adrese je stigao pa ona predstavlja asemblerov PC (program counter). Sadržaj joj možete promeniti na bilo kom mestu u programu pa različiti delovi mogu biti asemblirani na različite adrese. Znak jednako se generalno upotrebljava za dodelu vrednosti varijablama u programu. Specifičnost *= je u tome što te dva znaka predstavljaju pseudo naredbu pa se pišu bez razmaka — za sve ostale promenljive između naziva i znaka jednako obavezan je bar jedan razmak

Promenljive u asembleru

Čemu uopšte promenljiva u asembler programu? Druga upotreba je zamena nekih konstanti varijablom kojoj se vrednost dodeljuje u početku a zatim u celom toku programa koristi samo njen naziv. Na ovaj način se svaka izmena konstante u toku razvoja programa izvodi jednostavnom promenom vrednosti same varijable a ne menjanjem konkretne vrednosti na svim mestima u programu gde se pojavljuje. Nasamo smo rekli druga upotrebe jer se varijable primarno upotrebljavaju kao simbolička adrese. Svaki skok (JMP JSR ili Branch instrukcija) zahteva adresu na koju će biti izveden. Izračunavanje konkretne vrednosti se može prepustiti asembleru ako se ispred instrukcije na koju se skok vrši postavi simbolička adresa, ili labela, kako se često naziva

Primer programa koji će napuniti ekran kodovima od 32 do 128 i koji je već postao klasika za demonstriranje brzine mašinskog jezika dat je na slici 1 pod P2

To je otprilike sve što je potrebno znati da bi započeli rad sa asemblerom. Međutim, prava priča o asembleru počinje tek sada

Sistemi brojeva

Profi asembler raspoznaje tri sistema brojeva heksadecimalni koji ima prefiks $ binarni sa prefiksom % i decimalni bez prefiksa. Moguće je zadati negativne brojeve i ASCII format. Nekoliko pravilnih oblika je dato na slici 1 P3

Ni u jednom sistemu nije dozvoljeno prekoračenje maksimalne vrednosti koja se može predstaviti sa dva bajta ($FFFF ili 65535). Postoje i tri specijalna prefiksa sa sledećim značenjem

< — niži bajt navedene vrednosti

> — viši bajt navedene vrednosti

profi asembler 64

kako to radi

Programski jezik u koma svaka instrukcija mikroprocesora ima svoj mnemonički ekvivalent naziva se asembler. Jedino ovaj programski jezik omogućava da se iskoristi apsolutno sve za šta je jedan mikroprocesor sposoban, ali je cena često prevelika — da bi se napisao program potrebno je uložiti enorman trud. Programeri na malim mašinama, bez obzira na sve teškoće, još uvek najradije posežu za ovim jezikom. Za vlasnike „komodora" pripremili smo uputstva za „Profi. ass 64", koji se može dobiti i preko „Eprom-servisa „Računara". Do ovog programa smo došli sasvim slučajno. Nije nam poznat ni autor, ni kuća koja ga je izdala, a, kako to obično biva kod nas, niismo uspeli da dođemo ni do originalnog uputstva. Verujući da se radi o dobrom programu, Zoran Životić je, jednostavno, izlistao ceo program i na osnovu toga izveo sopstveno uputsvo, na kome je zasnovan i ovaj tekst.

I — forsiran dvobajtni format

Poslednji zahteva dodatno objašnjenje. Ako se u programu nađe instrukcija

```
LDA 100
```

asembler će je prevesti u 165 100, što je naredba LDA ZERO PAGE. Ako želite instrukciju LDA ABSOLUTE, čiji bi kod za dati slučaj bio 173 100 0, onda asembler možete naterati da broj 100 smatra dvobajtnim, bez obzira što je viši bajt 0, upotrebom LDA I100.

Adresa ili vrednost može biti, kako smo već videli, zadata i promenljivom ali i čitavim izrazom. Evo jednog „besmislenog" ali ilustrativnog primera koji će asembler sasvim korektno prevesti:

```
LDA #<(START+((BIT*16)!($FFFF ↑ FLG)+%01100000+ .A ) 3)
```

Aritmetičke i logičke operacije

Spisak dozvoljenih aritmetičkih i logičkih operacija dat je u tablici na slici 2. Dozvoljena je upotreba i običnih i uglastih zagrada — one imaju isti smisao pa izbor zavisi od vaših estetskih kriterija, ali se uglastim zagradama mogu izbeći nesporazumi jer se obične koriste i kao oznaka indirektnog adresiranja. Važno je znati da se izraz izračunava sleva nadesno bez ikakvih prioriteta operacija, osim onog uspostavljenog rasporedom zagrada. Sve operacije se obavljaju na standardan mikroprocesorski način, dakle po modulu 65536 pa će tako sabiranje, na primer 65000+636 dati rezultat O. Primećujete da se znaci ,< I I koriste i kao operatori i kao prefiksi. Zato je važno da znate da se kao prefiks odnose na CEO izraz i moraju se nalaziti na početku. Čak i ako u prošlom primeru izbacite prvu i poslednju zagradu, asemler će kao rezultat dati vrednost nižeg bajta celog izraza, a neće uzeti samo niži bajt varijable START i zatim izvršiti ostatak računanja.

Promena dodeljenih vrednosti

Posebna promenljiva čije se vrednost može slobodno koristiti je već pomenuta *. Njeno korišćenje je ravnopravno sa ostalim koje sami definišete, a vrednost joj je jednaka adresi na kojoj će se naći instrukcija koje je koristi. Na primer

```
LDA *+100
```

će, kada se prevede i startuje, smestiti u akumulator sadržaj memorijske lokacije udaljene 100 bajtova od same LDA instrukcije. Da bi ovo do kraja bilo jasno, reći ćemo da pojave labele u programu ima interni efekat:

```
LABELA=*
```

pa je gornji primer potpuno ekvivalentan sa

```
LABELA LDA LABELA+100
```

Dodela vrednosti promenljivima znakom = se može vršiti samo jednom u programu. Ovo je razumljivo, jer se na taj način sprečava pojava dve iste varijable. Ako je potrebno da se vrednost promeni, se to se koristi specijalan znak dodele, strelica levo

```
BROJAC=100
BROJAC ←BROJAC−1
```

Na prvi pogled, promena već dodeljene vrednosti nema smisla i mnogi asembleri je potpuno zabranjuju. Ipak, profi asembler poseduje nekoliko pseudo naredbi kojima mogućnost promene vrednosti varijabli znatno proširuje upotrebljivost.

Poruke i tablice

Tri pseudo naredbe, za razliku od ostalih, smeštaju neki kod u mašinski program kao da se radi o mikroprocesorskoj instrukciji. Ako je potrebno poruku ili tablicu reči smestiti u program, koristi se naredbe sa dovoljno ilustrativnim nazivom .ASC. Asembler će u trenutku nailaska na nju tekst između znakova navoda koji sledi direktno smestiti u memoriju i to od mesta do kog je stigao u prevođenju (sadržaj varijable *). Jedan jednostaven primer je dat na slici 1 P4. Upotrebili smo mali trik — zadali smo da se rezultujući kod smesti od adrese 1024, što je početak video memorije, pa je način na koji .ASC naredba smešta kod u memoriju vidljiv

SLIKA 1.

```
P1 100 PRINT"BEJZIK PROGRAM"
   110 SYS28672
   120 LDA #0
   130 STA 53281
   140 RTS

P2 100 SYS28672
   110 *= 10000
   120  OPT O0
   120 LIMIT = 130
   140 START = 32
   150 LDY #START
   160 LDX #0
   170 NEXTCHR  TYA
   180 LOOP     STA 1024,X
   190 INX
   200 BNE LOOP
   210 INY
   220 CPY #LIMIT
   230 BNE NEXTCHR
   240 RTS
   RUN
   SYS10000

P3 $FF02
   @A
   %01101
   34010110011011101
   23
   33287
   -12
   "A"
   itd.

P4 10 SYS28672
   20 *= 1024
   30 .OPT O0
   40 .ASC "TEKST........ . "

P5 10 SYS28672
   20 PASS = -1
   30 ..
   40 ;    PROGRAM 1.
   50 ..
   100 .PASS = PASS+1
   110 .IF PASS:.FIL 8,'naziv 2.'

P6 10 SYS28672
   20 .OPT O0
   30 *= 820
   40 WAIT LDA 653
   50 BNE WAIT
   60 RTS

P7 10 OPEN 2,8,4,"naziv,p,w"
   20 SYS28672
   30 *= početna adresa
   20 .OPT O2
   40 .....

P8 330 COUNT = 1
   340 .BYT 32
   350 .IF COUNT-200:COUNT = COU
   NT+1:.GOT 340
```

bemah u toku asembliranje. Samo da napomenemo da se između znakova navoda može naći bilo šta što komodor inače podržava — dakle kontrolni karakteri, grafički karakteri i sl.

Za pravljenje tablica brojeva služe naredbe BYT i WOR od kojih prva smešta po jedan bajt, a druga celu reč (dva bajta) u standardnom formatu niži—viši bajt. Dve stvari čine ove naredbe vrlo korisnim: (1) mogućnost da se više parametara, odvojenih zarezom, unese u okviru jedne naredbe i (2) se parametra važe sva pravila o prefiksima, izrazima, brojnom sistemu koji su ranije pomenuti. Tako je sasvim uobičajen neki od ovih oblika:

```
BYT <MSG1 <MSG2
BYT >MSG1 >MSG2
```

ili

```
WOR TEST+$FF,* — 16
```

Listanje programa

Ovaj nivo korišćenje asemblera završićemo još jednom opcijom OPT naredbe. Dokumentacija je jedna od važnih stvari pri radu u mašinskom jeziku. Bez obzira na kvalitet, bejzik editor se baš ne može pohvaliti preglednošću listinga. Kod asemblera je preglednost labela, adresa i instrukcija važna stavka, pa profi asembler poseduje svoj ispis. Ako u programu zadate OPT P na ekranu će biti ispisan listing programa, onako kako se prevodi. 40 znakova jedne linije je ipak malo da stanu svi podaci, pa je ispis razlomljen u dva reda i ponekad se teško snaći. OPT P je, zbog toga, prvenstveno namenjena za štampanje, ali se ovu priliku mora da ima i poseban oblik. Izabrano je elegantno rešenje koje omogućuje pravljenje bolične dokumentacije. Da bi izlistali program na printeru, treba da počnete ovako

```
10 OPEN 11 4,0
20 SYS 28672
30  OPT P11
```

Dakle asembler će izlistati program preko bilo koja veze koju ste otvorili u bejziku i čiji logički broj zadate (bez razmaka) uz slovo P Samim tim ispis se može uputiti i u neku datoteku na disku ili kasetofonu radi dalja obrade na primer u tekst procesoru Nakon upoznavanja se P opcijom vaši će programi najčešće započinjati naredbom

OPT OO P

Programi iz delova

Svaki složeniji program se lakše piše razbijano u više celina Profi asembler u potpunosti podržava ovakav red Potrebno je imati mogućnost da se vrednost verijabli pranosi iz jedna celine u drugu i automatiku u učitavanju pojedinih delova Prvi zadatak se može obaviti na više načina Na raspolaganju su dve naredbe

.SST dv,sa naziv" — snimi varijable

LST dv,sa naziv" — učitaj verijabla

pri čemu je dv broj uređaja koji se koristi kao spoljna memorija a se sekundarna adresa Kod ovih naredbi bi se mogla uputiti prva ozbiljnija zamerka asembleru Naime predviđen je rad preko serijskog interfejsa praktično samo sa diskom pa će dv uvek biti 8 sem kod retkih koji reda sa dva diska. Vlasnici kasetofona na moraju previše da se uzbuđuju Ima dovoljno načina da se ovaj posao ipak obavi Naredbe FIL i END će nešto dužim putem izvršiti ovaj zadatak. Metod je jednostavan Umesto da iz jedne celine snimite sve varijabla i zatim ih upisujete sa LST u svim drugim delovima, moraćete svaki put da propuštate sve delove kroz asembler ali će posao biti obavljen automatski pa se jedino može govoriti o gubitku vremena. Pretpostavimo da jedan veliki program treba razbiti na manje delove od kojih će prvi deo biti definisanje svih potrebnih konstanti u programu Na primer

```
10 SYS28672
   20 CHROUT=65490
   30 RANGE=65
   ITD
```

Ako ovaj deo završite naredbom

END dv naziv sledećeg dela"

asembler će proći kroz njega, interno formirati sve varijabla koje se pojavljuju upisati sledeći deo i asemblirati ga raspoznajući sve labele iz prvog dela Očigledno da bez diska posao na razvijanju recimo 4 dela može biti jako komplikovan pa se za rad sa kasetofonom ipak preporučuje što više smestiti u jednu celinu. Turbo tejpom se program može slobodno snimati ali END koristi isključivo C64 format što zahteva da delove pišete gotovo bez greške jer svaka ispravka i ponavljanja procesa asembliranja može da potraje

Još jedna naredba ima potpuno istu sintaksu i zadatak a to je FIL Male razlike ipak postoje Originalno uputstvo za profi asembler bi verovatno razjasnilo razlog postojanje oba Iz samog listinga programa to nije bilo moguće Jedina funkcionalna razlika je u tome što se FIL može izvesti u prvom i drugom prolazu asembliranja, dok END radi isključivo u prvom. Prvi i drugi prolaz?

Prvi i drugi prolaz(i)

Iako smo objašnjenje rada asemblera ostavili za kraj, ne znači da se radi o nečem komplikovanom Programi se mogu pisati i bez detaljnog poznavanje rada profi asemblera ali će to znanje omogućiti bolje korišćenje njegovih osobina

Da bi asemblirao program profi asembler mora da ga analizira dva puta. U prvom prolazu se samo proverava sintaksa naredbi i instrukcija (još se ne prevode u kodj i izvodi najvažniji deo — formiranje vrednosti varijabli i labela. Razlog je vrlo jasan Ako svaka verijabla treba da se koristi bilo gde u programu neophodno je imati sve njihove vrednosti u trenutku samog prevođenje instrukcija. U suprotnom bi bilo nemoguće obraćanje na primer skok na labelu unapred Kada se ovaj posao obavi u drugom prolazu se vrši prevođenje u mašinski kod Šta će se njim biti urađeno zavisi od izabranog parametra OPT naredba

Što se tiče pseudonaredbi one se izvode u zavisnosti od same instrukcija — neke u prvom, neke u drugom a najčešće u oba prolaza Poznavanje ove njihove osobine je presudno za razumevanje dejstva. Da bi to ilustrovali vratićemo se opet na END i FIL Ako se FIL izvodi u oba prolaza: šta će se desiti kada profi asembler naiđe na nju? Prvo to se dešava odmah u prvom prolazu a do tada su formirane samo varijable i to one koje se do tog trenutka pojavljuju FIL izvodi upisivanje sledećeg programa (dakle prvi deo se gubi) i profi asembler počinja njegovo asembliranje od početka Dakle posledica je pomalo neprijatna ako je niste očekivali Iz prvog dela će biti postavljene varijable ali ako se u njemu naša i neke instrukcija ona jednostavno neće biti prevedena (to se radi u drugom prolazu) Ujedno početna adresa za asembliranja drugog dela, ako je niste zadali očekujući da se kod nastavi na prethodni biće ista kao za prvi deo pa labela drugog dela neće imati mnogo smisla, jer program ne počinje od prave adresa END obavlja isti zadatak kao FIL ali samo u prvom prolazu dakle uvek na opisan način Očigledno je potrebno naći put ako je već dozvoljeno da se FIL izvede u drugom prolazu

Uslovno asembliranje

Postoji jedna važna osobina dodela vrednosti varijabli Ako se to radi znakom = onda se obavlja samo u prvom prolazu dok se promena vrednosti znakom dodele strelica levo obavlja u oba Nije dakle neizvodljivo da primenom nekoliko linija kao

```
150 PASS=1
160 PASS < PASS+1
```

GREŠKE U TOKU PREVOĐENJA PROGRAMA

0****SYNTAX	Nije dozvoljena samo labela bez instrukcije
1****SYNTAX	Nekorektan naziv labele
3****SYNTAX	U izrazu upotrebljen operator koji ne postoji
4****SYNTAX	Nekorektan raspored zagrada u izrazima
5****SYNTAX	Nekorektan prefiks (dozvoljene ! > $ %)
	Nekorektan parameter OPT naredbe (dozvoljeno O P N)
6****SYNTAX	Nekorektni parametri SST LST FIL END Parametri BYT WOR nisu odvojeni zarezom
7****SYNTAX	Nepostojeća pseudo-instrukcije
9****SYNTAX	Pogrešan način adresiranja. Upotrebljeni način adresiranja ne postoj na procesoru 6510
*ILLEGAL QUANTITY	Zadati broj izlazi van opsega (bajt ili reč)
*OVERFLOW	Linija predugačka, suviše parametara u BYT ili WOR
*BRANCH OUT OF RANGE	Relativni skok (branch instrukcija) van opsega)
*REDEFINITION	Pojava dve iste labele (varijable)
*UNDEF'D STATEMENT	Adresiranje na labelu koja nije definisana
*ISYM TABLE OVERFLOW	Tablica labela silazi ispod granice zadate sa STM
*!OUT OF MEMORY	Program i labele zahtevaju više memorije od raspoložive količine
*!UNDEF'D STATEMENT	GOT na nepostojeću liniju
*!DEVICE NOT PRESENT	Periferni uređaj nije priključen
*!IEEE	Greška u komunikaciji sa periferijom (ako STATUS <>0)
*!DISK ERROR	Greška na disk jedinici. Ispisuje se poruka sa ERROR kanala.

Pojava greške ne zaustavlja proces prevođenja ali mašinski kod u najvećem broju slučajeva nije korektan. Ako je greška u adresnom delu, kod instrukcije će biti pravilno izveden ali će adresa imati vrednost 0. Loše labele obično prouzrokuju više grešaka, sve instrukcije koje je koriste takođe proizvode grešku.

Poruku o grešci profi asembler ispisuje PRE linije na kojoj je nastala.

na osnovu stanja verijable PASS saznamo u kom prolazu je asembler Sad još samo mogućnost uslovnog izvođanja naredbi i Ako ste pomislili na neku konstrukciju tipa IF asevim ste u pravu Profi asembler zaista poseduje IF, ali se malo skraćenim oblikom ispituje se (u oba prolaza) da li je vrednost izraza jednaka 0 ili ne Dakle nema poređenja tipa veće—manje i ostalih kombinacija Namerno smo rekli izraza a ne varijabla jer za vrednost koja se ispituje važe svi mogući oblici numeričkih logičkih i ostalih izraza koje dopušta profi asembler Ako je vrednost različita od nule biće izvedene instrukcija iza IF u suprotnom se preskače osta linija. Sintaksa IF ne zahteva upotrebu THEN potrebno je samo

IF izraz naredba naredba

Dve tačke iza izraza nisu nikakva specifičnost ove konstrukcija Profi asembler dozvoljava kao i bejzik odvajanja naredbi ili instrukcija ovim znakom, pa u jednu liniju može stati dosta toga Iz iskustva vam savetujemo da ovo ne zloupotrebljavate bar ne u početku jer sigurno vodi potpunoj nepreglednosti i teškom snalaženju pri ispravkama programa. Dovoljno će biti da se ograničite na jednu ili dva instrukcija u jednom redu Dakle deo koji će obezbediti potpuno povezivanje (i labela i koda) više programa treba da izgleda kao na slici 1 PS

Ako umesto FIL upotrebite END neće se desiti ništa jer END radi samo u prvom prolazu U drugom prolazu će imati isti efekat kao i END bez parametara kojim se završava asembler program i vraća kontrola bejziku

Pseudonaredba IF ne bi bila kompletna bez GOTO što je bilo jasno i autorima PA Zato postoji GOT naredba koja kao parametar zahteva linijski broj na koji od tog trenutka treba preći sa asembliranjem GOT se izvodi u oba prolaza, o čemu treba posebno voditi računa Prva logična upotreba je u fazi izrade kada treba neki deo asembler programa privremeno izbaciti Da ga ne bi brisali pa kasnije ponovo upisivali ispred dela zadajte GOT na prvu liniju iza njega i to je sve Daleko je interesantnija kombinacija IF GOT kojom se mogu lako ostvariti složeni zadaci Pitanja Kako napraviti tablicu od 200 bajtova koji inicijalno treba da imaju vrednost 32? Odgovor Upotrebiti dve stotine BYT 32 instrukcija se ne priznaje zbog nehumanog odnosa prema programeru Ostaje elegantno rešenja sa slika 1 P6 U primeru su značajne dve stvari Prvo upotrebljena je dodela vrednosti znakom strelica levo Tako će COUNT u liniji 330 dobiti vrednost 1 i pri drugom prolazu Da je upotrebljeno = desilo bi se da nakon 350 linija kada vrednost dostigne 200 varijabla COUNT ostane na toj vrednosti i u drugom prolazu uslov za ponavljanje BYT instrukcija već u startu ne bude ispunjen pa samim tim bajt 32 bude smešten u memoriju samo jednom Treba takođe napomenuti da u tom slučaju ni eventualne labela nakon ovakvog dela ne bi u drugom prolazu bile korektne jer im je vrednost formirana kao da će dva stotine bajtova

NAREDBE I SINTAKSE PROFI-ASS. 64.

LABELE DO 8 znakova prvo slovo
zatim slovo ili broj
standardna labela PC

PREFIKSI OPERATORI I OZNAKE

$	heksadecimalni		niži bajt	+	sabiranje
	binarni		viši bajt		oduzimanje
	ASCII	!	forsiran dvo-		množenje
	decimalni		bajtni format	&	AND
				!	OR
	komentar				EOR
	dodela vrednosti va[illegible]				rotiranje [illegible]
←	dodela i promena [illegible]				rotiranje [illegible]

naredba	prolaz	sintaksa	komentar
*=	1 2	* adre[illegible]	dodela vrednosti P[illegible]
BYT	2	BYT bajt bajt	tablica bajtova
WOR	2	WOR reč reč	tablica reči (dva bajta)
ASC		A C asci n z	tablica ASCII karaktera
OPT		OPT O	smešta kod od adrese STREND (49)
OPT	2	OPT OO	smešta kod od adrese *
OPT	2	OPT On	smešta kod u datoteku sa logič brojem n
OPT	2	OPT O adresa	za svaku instrukciju poziva maš p na adre[illegible]
OPT	2	OPT P	ispis listinga na ekran
OPT	2	OPT Pn	ispis u datoteku sa logičkim brojem n
OPT	2	OPT P adresa	pre ispisa svakog karaktera poziva mašinski progra[illegible]
OPT	2	OPT N	poništava sve ranije OPT
OPT		OPT N OO P	dozvoljeno više parametara odvojenih zarezom
END	2	END	kraj asembler programa
END		END dv naziv	kraj programa i upis sledećeg
FIL	2	FIL dv naziv	upis sledećeg asembler programa
SST		SST dv sa naziv	snima varijable (simboličke adrese)
LST		LST dv sa naziv	učitava varijable (simboličke adrese)
STM	2	STM adresa	ograničenje prostora za varijable do adrese n
SYS	2	SYS adresa	poziv mašinskog programa na adresi
IF	2	IF izraz naredbe	ako izraz O izvedi naredbe
GOT	2	GOT n	asembliranje prenesi na liniju n
HTB	1 2	H B	izlazak iz asembliranja preko BWS vektora (bejzik warm start)

zaista biti u memoriji, dok će stvarno smeštanje koda početi odmah nakon samo jednog bajta 32. Druga stvar je ispitivanje dostignute granice. Pošto IF registruje samo dva slučaja jednako ili različito od nula, potrebno je izraz tako svesti da se uslov može ispitati ovim vrednostima. U primeru je to jednostavno izvedeno oduzimanjem 200, ali ponekad se ceo izraz može dosta iskomplikovati. Ne treba zaboraviti da vrlo korisno mogu da posluže logičke i shift operacije, ali tada morate da „razmišljate binarno".

Ako ste u stanju da primenite sve do sada rečeno, neće vam biti teško da shvatite ostatak naredbi iz tablice koju dajemo u prilog, a koje ovde nismo posebno objasnili. Ostaje nam samo da vas upoznamo sa nekim tehničkim detaljima koji mogu olakšati rad.

Finese naredbe .OPT

Treba znati da profi asembler koristi 35 bajtova memorijskog prostora iz sistemskog dela od adrese 57. Sadržaj ovih lokacija se privremeno čuva u INPUT baferu (od 512). Ako asembliranje prekinete sa STOP+RESTORE (što inače nije neophodno, profi asembler podržava prekid samo STOP tasterom) originalne vrednosti neće biti vraćene pa izvođenje neke od bejzik funkcija, na primer PEEK, može blokirati računar. Tada vam ostaje da odmah snimite program, inicijalizujete računar i tek onda nastavite rad.

Varijable se smeštaju od najviše slobodne adrese naniže, kao što to bejzik radi sa stringovima. One formirane u asembleru se ne raspoznaju u bejziku, a na žalost, isto važi i obratno.

Slika 3

```
10 REM ** ISPRAVKA PROFI-ASS 64 **
20 RESTORE:FOR F=29762 TO 29809
30 READB POKEF,B:NEXTF
40 DATA 6,76,38,77,176,48,101,76
50 DATA 170,169,0,101,77,176,39,72
60 DATA 138,72,162,2,6,76,38,77
70 DATA 176,28,202,208,247,104,101,76
80 DATA 133,76,104,101,77,176,15,133
90 DATA 77,234,234,234,234,234,234,234
```

Spisak svih parametara OPT naredbe je dat u tablici, ali oblik OPT P=adresa ili OPT O=adresa zahteva dodatno objašnjenje. Radi se o tome da su autori profi asemblera, što je vrlo korektan potez, ostavili mogućnost da se vašim programom ubacite u proceduru ispisa ili asembliranja. Na primer, listanje izazvano opcijom P se odvija na standardan C64 način, dakle brzo uz malu mogućnost usporavanja CTRL tasterom. Za analizu listinga ovo je i dalje prebrzo, pa se može upotrebiti sledeći metod. Kreirajte mali program (slika 1, P6). Asemblirajte ga, obrišite i počnite sa radom na vašem programu, ali za ispis upotrebite opciju OPT P=820.

Sada će profi asembler pre svakog ispisanog znaka izvesti JSR 820 i time pozvati malopre kreirani program. Njegov zadatak je da se vrti u praznoj petlji ako je pritisnut neki od tastera SHIFT, C= ili CTRL, čime se listanje potpuno zaustavlja. Pri ulasku u ovu rutinu u akumulatoru se nalazi bajt koji se ispisuje, što ovde nije iskorišćeno, ali se može upotrebiti za posebna usmeravanja ispise i slično.

Mehanizam koji se primenjuje na O=adresa se razlikuje utoliko što se skok na adresu vrši pre smeštanja celog koda jedne instrukcije i potpuno se preskače smeštanje od strane profi asemblera. Broj bajtova instrukcije se nalazi u lokaciji 78, sam kod počinje od 75, ali ako je duži od tri bajta (eventualno kod WOR, BYT i .ASC) ostatak počinje od adrese 347. Nije baš jednostavno ali se retko i koristi. Prvo što padne na pamet bi bilo relocirano asembliranje, ali je i ono podržano od strane PA pa ova opcija uglavnom završava na zadovoljstvu što postoji ako ikad zatreba.

Relocirano asembliranje se izvodi opcijom OPT O. Navođenje početne adrese se i dalje obavlja sa *= i program će biti asembliran kao da treba da radi od nje, ali će sam kod biti smešten od adrese na koju pokazuje sistemska varijabla bejzika STREND (49 50). Ako koristite disk, elegantnije rešenje omogućava opcija OPT On, gde je n logički broj veze otvorene u bejziku. Efekat je sličan opciji OPT Pn, ali se sada ne usmerava listing u datoteku već sam rezultujući mašinski program (slika 1, P7). Na ovaj način se na disku dobija gotov program koji jednostavno treba upisivati sa LOAD "naziv",8,1. Početna adresa može imati bilo koju vrednost jer program ne dodiruje memoriju u fazi asembliranja. Na žalost, ovako kreirane datoteke je sekvencijalnog tipa (a ne program), pa iako se na disku uz otvaranje sa ,P,W lako prevodi u programsku, na kasetofonu je to nemoguće i kreirana datoteka neće moći da se upisuje naredbom LOAD.

Verujemo da su vam sada potpuno jasni razlozi zbog kojih ovaj program smatramo odličnim. Treba dodati i to da su autori uložili dosta truda da omoguće i neke sitne detalje kao što je upotreba bejzik službenih reči kao varijabli. Pošto se pri unošenju linije odmah vrši tokenizacija, na primer PRINT labela će biti predstavljena jednim bajtom, a ne nizom slova P R I N T, ipak profi asembler će se ponašati sasvim korektno, pa je upotreba svakog naziva dozvoljena, čak i pseudo naredbi ako ne počinju tačkom, na primer OPT=100.

Nije korektno prema autorima programa da na samom kraju spominjemo ozbiljnu grešku koju su napravili, ali nas tok ovog teksta na to prisiljava. Pri prevođenju niza decimalnih cifara u binarni broj na jednom mestu je zaboravljeno resetovanje CARRY zastavice, pa se određeni brojevi prevode na vrednost za jedan veću od stvarne. U verziji prilagođenoj za kartridž kojim redakcija programira EPROME ova greška je ispravljena, ali možete isto uraditi i sa standardnom verzijom primenom programa sa slike 3. Upišite asembler, ukucajte i startujte dati program, zatim snimite ispravnu verziju profi asemblera.

Zoran Životić

Matematički softver

To može i bolje (7)

arkus funkcije

Ovde su proučavane mogućnosti upotrebe pojedinih formula za izračunavanje arkus funkcija koje se preporučuju u literaturi ili su ugrađene u osnovnu programsku podršku računara. Dvanaest priloženih crteža nedvosmisleno pokazuju da se istraživanju valjanih algoritama za izračunavanje vrednosti matematičkih funkcija nije poklanjala potrebna pažnje Uz analizu kako ne treba i zašto tako ne treba programirati ove funkcije, priloženi su programi koji pokazuju kako se mogu valjano izračunavati vrednosti arkus funkcija Na apscisi priloženih crteža uvek je u pogodnoj razmeri argument, a na ordinati je tačnost izračunavanja merena brojem tačnih bitova mantise Iako je razmatranje opštije, programi su napisani za računar koji ime 32 bita mantise (m=32), na primer „šarp MZ-700". Za „komodor 64" i „spektrum 48" u programe treba na odgovarajućim mestima ubaciti reči THEN i LET

Arkussinus

Funkcija arkussinus arcsin(x) je definisana za abs(x)<=1 Funkcija arcsin(x) je rastuća od $-\pi/2$ do $\pi/2$ Za mala vrednosti modula argumenta abs(x) može se uspešno aproksimirati formulom

$$(1)\ \arcsin(x) \sum_{k=0}^{n} (2K)!x^{2K+1}/2^{2K}K!^2(2K+1)$$

Za razne vrednosti gornje granice sumiranja tačnost formula (1) predstavljena je na slici 1 Slika 1 pokazuje da je za 32 binarne cifre mantise i za argumente po modulu manje od 1/2 potrebno uzeti n=12 Primenom ekonomizacije ovaj broj n može se smanjiti uz izmenu vrednosti koeficijenata ali to je posebna tema Ako se nekome ne čini da je 14 množenja i sabiranja mnogo za 32 binarne cifre mantise (skoro 10 decimalne cifre) za abs(x)<=1/2 može koristiti formulu (1) sa n=12

Naravno, ostaje teži deo problema kako računati arcsin(x) za 1/2<abs(x)<=1 Pri krajevima intervala definisanosti izvod funkcije neograničeno raste pa treba očekivati da je u tim oblastima izračunavanje vrednosti funkcija otežano

Poznatom formulom

$$(2)\quad \arcsin(x)=\pi/2-\arcsin(sqr(1-x^2))$$

moguće je izračunati arcsin(x) za sqr(1/2)<x<=1 ako se zna ta funkcija za abs(x)<=sqr(1/2). To bi značilo da ranija aproksimacija (1) za n=12 nije dovoljna, već treba povećati n. Greška formula (2) predstavljena je na slici 2 gde je uočljivo da se skoro četvrtinu broja bitova gube na kvadriranje argumenta i oduzimanja tog kvadrata od jedinice Povećanje greška

```
10 LABEL "ARCSIN"
11 IF ABS(X)<.000027727035900 Y=X:RETURN
12 Z=X
13 Y=SQR((.5-X+.5)*(.5+X+.5))+1D-30
14 X=X/Y
15 GOSUB 31
16 X=Z
17 RETURN

20 LABEL "ARCCOS"
21 IF ABS(X)< 00140044005#
   Y=1.570796326795#-X RETURN
22 Z=X
23 X=SQR((.5-X+.5)/(.5+X+.5+1D-30))
24 GOSUB 31
25 Y=Y+Y
26 X=Z
27 RETURN

30 LABEL "ARCTG"
31 Y=ABS(X)
32 IF Y< ##00277270530#Y=X  RETURN
33 IF Y>2147483648#
   Y=1.570796326795# GOTO 41
34 IF Y< 23052030570 U=0:V=0:GOTO 38
35 IF Y< 0001440515# U=.453125:
   V= 4088150035430 GOTO 38
36 IF Y<2.1352520### U=.50025:
   V=1 2704010205020  GOTO 38
37 U=1 355375:V=4.055207120
38 U=(Y-U)/(Y*U+1)
39 W=-U*U
40 Y=(((((.0730*W+,09050505)*W+
   .1111111111#)*W+ 1428571425#)*W+
   2)*W+,3333333333#)*W*U+U+U
41 IF X<0 Y=-Y
42 RETURN
```

izračunavanja preko sto puta je, naravno nedopustivo

Sledeća formula

$$(3)\arcsin(x)=\pi/2-2\ \arcsin(sqr((1-x)/2))$$

omogućava izračunavanje funkcije arcsin(x) za 1/2<x<=1 ako je funkcija poznate za 0<=x<1/2 Na slici 3 data je greška formula (3) koja potiče od nebrižljivog oduzimanja 1−x Ovo oduzimanja se more obaviti brižljivija, npr sa 0 5−x+0 5 Na taj način moguće je izračunati funkciju arcsin(x) za 0<=x<=1 Za −1<=x<0 funkcija se lako izračunava, jer je neparna

$$\arcsin(x)=-\arcsin(-x).$$

Rasprostranjena je zabluda da se arcsin(x) može računati pomoću arccos(x) formulom

$$(4)\qquad \arcsin(x)=\pi/2-\arccos(x),$$

naravno pod pretpostavkom da se funkcija arccos(x) valjano računa. Da je to zabluda upečatljivo pokazuje slika 4,jer je uočljivo da se za mala vrednosti modula argumenta ne dobija nijedna tačna cifre rezultata. To je jedan od brojnih primera analitički tačne formule koje je numerički neupotrebljiva

Računarski programi najčešće sadrže formulu

$$\arcsin(x)=arctg(x/sqr(1-x^2))$$

Greška ove formule je kao na slici 2, one pretežno potiče od kvadriranja argumenta i od oduzimanje tog kvadrata od jedinice Poslednja formula nije definisana za abs(x)=1, jer nije dozvoljeno deliti nulom Za x=1 ili x=−1 treba uzeti da je

$$\arcsin(x)=(\pi/2)sgn(x)$$

U „spektrum 48" firma je ugradila formulu

$$\arcsin(x)=2arctg(x/(1+sqr(1+sqr(1-x^2)))$$

koja ima približno istu grešku kao formula (2) Kod te formule nema opasnosti od deljanja nulom za x=1 ili x=−1, ali je ona složenija od prethodne formula

Sledećom formulom pokušava se da se smanji greška izračunavanja argumenta korena

$$\arcsin(x)=arctg(x/sqr((1-x)(1+x)))$$

greška je kao na slici 4 Treba imati u vidu da se čak do polovine bitova mantise izgubi samo zbog greška na poslednjem bitu u oduzimanju 1−x ili sabiranju 1+x Da se smanji ogromna greška argumenta korena može se (1−x)(1+x) napisati tananije sa (.5−x+.5)(.5+x+.5) Da se izbegne testiranja argumenta x na 1 ili na −1, dovoljno je korenu fiktivno dodati 1E−36 Ovo sabiranja obično se ne vrši zbog velika razlika u vrednosti binarnih karakteristika sabiraka pa je veoma brzo Sabiranja se vrši samo za x=1 ili x=−1 tada je zbir matematički gledano pogrešan — ali šta to mari kad je krajnji rezultat tačan

Prema tome funkciju arkussinus treba računati formulom

$$\arcsin(x)=arctg(x/(sqr((.5-x+.5)(.5+x+.5))+1E-36))$$

Za mala vrednosti modula argumenta kada je $x^3/6$ zanemarljivo u odnosu na x, treba koristiti formulu arcsin(x)=x

Za abs(x)>1 trebalo bi javiti grešku To nije naročito potrebno urediti iz ovog programa, jer u tom slučaju program za izračunavanje kvadratnog korena javija grešku

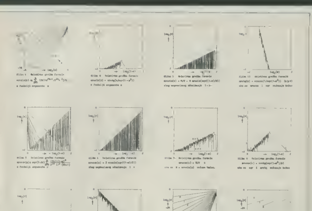
Arkuskosinus

Funkcija arkuskosinus arccos(x) definisana je za abs(x)<=1 Funkcija arccos(x) je opadajuće od π do 0, nije ni parna ni neparna Kako važi formula

$$\arccos(x)=\pi-\arccos(-x)$$

sleduje da se za negativne argumente lako računa ako je poznata za pozitivne

Arkus funkcije arcsin(x), arccos(x), arctg(x) pripadaju matematičkom softveru računara. Teško je naći računar kod kojeg su ove funkcije predstavljene korektnim programima. U seriji „To može i bolje" prof. dr Dušan Slavić izlaže detalje svog dvadeset godina dugog istraživanja u numeričkoj matematici.



Uopšteni potencijalni razvoj

(5)

$$\arccos(x)=\mathrm{sqr}(2-2x)\sum_{k=0}^{n}(2k)!(1-x)^k/2^{3k}k!^2(2k+1)$$

pokazuje se kao numerički nestabilan vidati sliku 5. Kada x raste ka 1 tačnost opada na nula tačnih cifara mantise. To što analitička greška sa porastom broja sabiraka opada slaba je uteha. Ukupna greška je zbir analitičke i numeričke, a ona čini da formulu (5) moramo proglasiti za numerički neupotrebljivu.

Iz formula (1) i (5) sleduje formula

(6) arccos(x)=2arcsin(sqr((1−x)/2))

Pod pretpostavkom da su programi za kvadratni koren i arkussinus idealno dobro načinjeni, greška formule (6) predstavljena je na slici 6. Greška potiče od nebrižljivog oduzimanja 1−x.

Kod većina računara izračunavanje funkcija arccos(x) svodi se na izračunavanja funkcije arcsin(x) pomoću formula

(7) arccos(x)=π/2−arcsin(x)

Da je taj jednostavan metod pogrešan vidi se na slici 7. kada x raste od 0 ka 1 broj tačnih bitova mantise opada od J na J/2, izgubiti polovinu tačnih cifara zbog jednostavne formule — to nije naročito mudro.

Kod nekih velikih računara, za 0<x<=1 koristi se formula

(8) $\arccos(x)=\mathrm{arctg}(\mathrm{sqr}(1-x^2)/x)$

Pod pretpostavkom da se funkcije kvadratni koren i arkustangens idealno tačno računaju greška formule (8) data je na slici 8. Greška potiče od kvadriranja argumenta i od oduzimanja tog kvadrata od jedinice.

Pa kako onda treba računati arkuskosinus? Treba, na primer, pomoću formule (6) i (8) dobiti formulu

arccos(x)=2arctg((1−x)/(1+x))
(−1<x<=1)

da se pomoću sličnog eksperimenta uveriti da ni one (iako brža od prethodnih) nije imune od numeričke greške

Zatim treba nepromišljeni izraz $(1-x)/(1+x)$ zameniti boljim $(.5-x+.5)/(.5+x+.5)$, čime se radikalno smanjuje greška

Uslov da x nije −1 može se eliminisati ako se imeniocu doda neki mali broj, npr 1E−36

Valjana formula postaje

arccos(x)=2arctg(sqr((.5−x+.5)/(.5+x+.5+1E−36)))

U okolini tačke x=0 celishodno je usvojiti aproksimaciju arccos(x)=π/2−x a one važi sve dok je $x^3/6$ zanemarljivo u odnosu na π/2 Tim testom se neznatno usporava program ako uslov nije ispunjen ali se znatno ubrzava program ako je uslov ispunjen

Arkustangens

Funkcija arkustangens definisana je za sve realne vrednosti argumenta. Ona raste od −π/2 (za x teži minus beskonačno) do π/2 (za x teži plus beskonačno) Za velike vrednosti modula argumenta veće od recipročne vrednosti elementarne promene mantise može se usvojiti aproksimacija

arctg(x)=(π/2)sgn(x)

Funkcija arkustangens je neparna

arctg(x)=−arctg(−x)

pa je jednostavno vrednost funkcije negativnog argumenta svesti na vrednost funkcije pozitivnog argumenta

Za mala vrednosti modula argumenta ako je $x^3/3$ zanemarljivo u odnosu na x moguće je arctg(x) aproksimirati sa x Time se ako je uslov ispunjen znatno ubrzava izvršavanja programa a ako uslov nije ispunjen neznatno se usporava

Gledano analitički moguće je funkciju arkustangens izračunati pomoću funkcije arkussinus npr sa

(9) arctg(x)=arcsin(x/sqr(1+x^2))

Da to nije preporučljivo dokaz je slike 9 pod pretpostavkom da su programi za arkussinus i kvadratni koren idealno tačni formulom (9) gubi se i do polovine binarnih cifara mantise Najveća greška je za abs(x) $2^{1/2}$ Može se reći da je arkustangens moguće računati pomoću valjanih programa za arkussinus i kvadratni koren samo za abs(x)<=1

Za x>1 vrednost arctg(x) može se izračunati pomoću vrednosti arkustangensa za argumente od 0 do 1 pomoću formule

arctg(x)=π/2−arcrg(1/x)

Uz neparnost funkcije arctg(x), rečeno je dovoljno za svođenje funkcije arkustangens na proste aproksimacije i funkcije arkussinus i kvadratni koren Ispitivanje numeričke greške pokazuje da je ipak bolje izračunavanje vrednosti funkcije arkussinus svesti na arkustangens nego obrnuto

Pod pretpostavkom da se funkcije arkuskosinus i kvadratni koren idealno tačno računaju na slici 10 predstavljena je tačnost formule

(10) arctg(x)=arccos(1/sqr(1+x^2)) (x>=0)

Slika pokazuje da je formula (10) izrazito loša za abs(x)<1 za abs(x)<$2^{1/2}$ ne dobije se nijadna tačne cifra rezultata. Formule (10) bi se mogla koristiti za računanje arctg(x) samo za abs(x)>1 Nevolje je jedino u tome što se (bar do sada) valjan program za arccos(x) ne može načiniti bez funkcije arctg(x) a ne obrnuto

Očigledno funkciju arkustangens treba računati bez pozivanja funkcija arkussinus arkuskosinus ili kvadratni koren

Na slici 11 data je tačnost formula

(11)

$$\mathrm{arctg}(x) \sum_{k=a}^{n} (-1)^k x^{2k+1}/(2k+1) \quad (\mathrm{abs}(x)<=1)$$

za razne vrednosti gornje granice sumiranja n Za zadani interval primene formula i zadanu tačnost računara sa slike 11 čita se koliko treba uzeti sabiraka u razvoju Ako najveća dozvoljena vrednost za abs(x) teži 1 broj sabiraka neograničeno raste Dakle, potencijalni razvoj (11) može se koristiti samo za vrednosti abs(x) manje od 1

Na slici 12 predstavljena je tačnost formule

(12)

arctg(x)=x/(1+1^2x^2/(3+2^2x^2/(5+ m^2 x^2/(2m+1))))

koje je posebno dobra za veće tačnosti računara

U literaturi se često funkcija arctg(x) aproksimira samo za abs(x)<=sqr(1/3) jer važi formula

arctg(x)=π/3+arctg((x−sqr(3))/(sqr(3)x+1))

gde je x>sqr(3)

Uz neparnost funkcije arctg(x) i pomenute aproksimacije poslednje formula je dovoljna. U tom slučaju je n=13 odnosno m=7 Dakle ako se želi brže izračunavanje ne sme se interval funkcije (−π/2 π/2) deliti samo na tri podintervala Za abs(x) <=tan(29/128) i J=32 sa slika 11 i 12 dobije se n=6 i m=5, što odgovara podeli intervala funkcije na sedam podintervale Naravno, ekonomizacijom se može uštedeti neka operacija množenja i sabiranja

Svođenje proizvoljnog argumenta na osnovni interval obavlje se formulom

arctg(x)=arctg(a)+arctg((x−a)/(ax+1))

Program sadrži izračunate vrednosti za arctg(a) i a Ako je broj tih izračunatih vrednosti veliki tj broj podintervala veliki onda je izračunavanje arctg(x) brzo — jedino je konstatovanje u kom se podintervalu nalazi argument x dugotrajno to znači da ni se brojem podintervala ne treba preterivati Ako je broj podintervala mali lako je svođenje na osnovni interval — ali je izračunavanje arkustangensa dugotrajno, ne treba preterivati ni sa štednjom izračunatih vrednosti za arctg(a) i a Ova protivurečna situacija rešava se u skladu sa opredeljenjem da li treba više štedeti na memoriji računara ili na brzini rada programa Za tačnost od 32 binarne cifre mantise (skoro 10 značajnih cifara rezultata) celishodno je usvojiti da se interval funkcije (π/2 π/2) deli na sedam podintervala Za veće tačnosti računara trebalo bi da broj podintervala bude veći

U priloženim programima provedeni su rezultati ovde datih razmatranja Funkcije arkussinus i arkuscosinus izračunavaju se pomoću funkcije arkustangens Za vrednost arkustangensa u osnovnom intervalu primenjena je ekonomizacija, koje će biti obrađena drugom prilikom

Dušan Slavić

Numerički metodi (3)

interpolacioni polinomi

U prethodnim nastavcima „Numeričkih metoda" razmatran je zadatak interpolacije u kome na osnovu poznatih

$x_0, x_1, x_2, \ldots, x_n$ (1)

$f(x_0)=y_0,\ f(x_1)=y_1,\ \ldots,\ f(x_n)=y_n$ (2)

najpre treba odrediti neku funkciju F(x) koja zadovoljava uslov

$f(x_0)=F(x_0),\ \ldots\ f(x_n)=F(x_n)$ (3)

a zatim traženu nepoznatu vrednost f(x) za zadato x izračunati kao F(x) Opisana je linearna interpolacija (F je linearna funkcija) uveden opšti pristup polinomijalnoj interpolaciji (F je polinom) i dat metod, program i procena greška za Legranžovu interpolaciju

Njutnov interpolacioni polinom

Kao i u slučaju Legranžovog interpolacionog polinoma i u Njutnovoj metodi se polinom zapisuje u pogodnom obliku koji omogućava neposredno dobijanje koeficijenata bez potrebe da se svaki put rešava pridruženi sistem jednačina Neka su tačke (1) na jednakom rastojanju h Njutnov polinom ima oblik

$$y(x_0+th)=y_0+\frac{t}{1!}\Delta y_0+\frac{t(t-1)}{2!}\Delta^2 y_0+\ldots+\frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_0 \quad (4)$$

gde su tzv konačne razlike uvedene sa

$$\Delta y_0=y_1-y_0,\ \Delta y_1=y_2-y_1, \ldots$$
$$\Delta^2 y_0=\Delta y_1-\Delta y_0,\ \Delta^2 y_1=\Delta y_2-\Delta y_1, \ldots \quad (5)$$

i tako dalje a nova promenljiva t definisana vezom

$$x=x_0+th \leftrightarrow t=\frac{x-x_0}{h}$$

Čitaoci će lako sami napisati program koji izračunava vrednosti polinoma (4)

Aitkenov metod

Navodimo program za polinomijalnu interpolaciju koji se zasniva na Aitkenovom postupku U ovom postupku se primenjuje ideja iterirane interpolacije, odnosno uzastopnog interpoliranja, pri čemu se tačnost uvećava do tražene (ako je moguće) Pro-

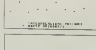
Mnogi konkretni problemi koji se rešavaju na računarima imaju u svojoj osnovi zadatke numeričke matematike. Numerički metodi poseduju veliku opštost i primenljivost, ali kako njihovo razumevanje i programiranje nije uvek sasvim jednostavno, to im upotreba, odnosno popularnost među vlasnicima mikroračunara, nije na nivou koji zaslužuju. Ova serija napisa ima za cilj da jednostavno, i samo u najvažnijim elementima, prikaže nekoliko metoda kojima se rešavaju neki osnovni zadaci numeričke matematike, kao i da ponudi programe koji su bazirani na tim metodama. Programi su pisani na bejziku za „komdor 64", ali se jednostavno mogu preneti na druge mikroračunare.

```
100 REM ****************************************
105 REM *                                      *
110 REM *            INTERPOLACIJA             *
115 REM *                                      *
120 REM *        METOD AITKEN-LAGRANGE         *
125 REM *                                      *
130 REM ****************************************
135 REM *          OPIS PROMENLJIVIH           *
140 REM ****************************************
145 REM *                                      *
150 REM * X-VREDNOST ARGUMENTA INTERPOLACIJE   *
155 REM *   (ULAZ)                             *
160 REM * A(I)-NIZ VREDNOSTI NEZAVISNO PROMENLJIVE *
165 REM *   (ULAZ)                             *
170 REM * A-PREUREDJEN NIZ A                   *
175 REM * V(I)-ULAZNI NIZ VREDNOSTI FUNKCIJE   *
180 REM * V PREUREDJEN NIZ V                   *
185 REM * B-BROJ TACAKA U NIZOVIMA A I V (ULAZ) *
190 REM * T-ZADATA GRANICA APSOLUTNE GRESKE    *
195 REM *   (ULAZ)                             *
200 REM * Y-INTERPOLIRANA VREDNOST FUNKCIJE    *
205 REM *   (IZLAZ)                            *
210 REM *                                      *
215 REM ****************************************
220 PRINT CHR$(147)
225 PRINT" ***********************************"
230 PRINT" * PROGRAM ZA INTERPOLACIJU FUNKCIJE *"
235 PRINT" *    JEDNE NEZAVISNO PROMENLJIVE    *"
240 PRINT" *                                   *"
245 PRINT" *       METOD AITKEN-LAGRANGE       *"
250 PRINT" ***********************************"
255 NI=8
260 GOSUB775:REM IZDAVANJE PRAZNIH REDOVA
265 INPUT"  PRITISNITE RETURN DA NASTAVITE ";PR
270 PRINT CHR$(147)
275 DIM A(100),V(100),X(100),V(100),N(100)
280 REM ****************************************
285 REM *             ULAZNI PODACI            *
290 REM ****************************************
295 INPUT "UNESITE BROJ TACAKA";B:PRINT
300 PRINT"UNESITE JEDAN IZA DRUGOG PAROVE"
305 PRINT"TACAKA  X(I),Y(I) ":PRINT
310 FOR L=1 TO B
315 PRINT"   X(";L;") Y(";L;") =";
320 INPUT A(L),V(L)
325 NEXT L
330 PRINT
335 PRINT"AKO JE I-TI RED POGRESAN, UNESITE I"
340 INPUT"AKO NEMA GRESKE UNESITE 0";N
345 PRINT
350 IF N=0 THEN 380:REM TACNO UNETO, DALJE
355 IF N>B THEN 330:REM VECE OD DIMENZIJE, PONOVI
```

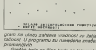
```
360 PRINT"UNESITE X(";N;"),Y(";N;") =";
365 INPUT A(N),V(N):REM UNOSENJE ISPRAVKE
370 GOTO 330
375 PRINT
380 INPUT "UNESITE TACNOST KOJU TRAZITE";T:PRINT
385 INPUT "UNESITE ARGUMENT INTERPOLACIJE";X
390 GOSUB 830:REM PREUREDJENJE A I V
395 PRINT:PRINT
400 PRINT"******  UNOS PODATAKA ZAVRSEN ******"
405 D=3:D2=0
410 IF B<1 THEN 585:REM DIMENZIJA 0
415 IF B=1 THEN 515:REM DIMENZIJA 1
420 REM ****************************************
425 REM *     POCETAK CIKLUSA ZA ITERACIJE     *
430 REM ****************************************
435 FOR J=2 TO B
440 D1=D2:K=J-1
445 FOR I=1 TO K
450 H=A(I)-A(J)
455 IF H=0 THEN 620
460 V(J)=(V(I)*(X-A(J))-V(J)*(X-A(I)))/H
465 NEXT I
470 D2=ABS(V(J)-V(K))
475 IF J<=2 THEN 495
480 IF D2 <= T THEN 570
485 IF J<5 THEN 495
490 IF D2 >= D1 THEN 590
495 NEXT J
500 REM ****************************************
505 REM *          ZAVRSETAK CIKLUSA           *
510 REM ****************************************
515 J=B
520 Y=V(J)
525 GOSUB 640:REM IZDAVANJE REZULTATA
530 NI=8
535 GOSUB 775
540 INPUT"(1)-NOVA INTERPOLACIJA, (2)-KRAJ";N
545 IF N >1 THEN END
550 GOTO 375
555 REM ****************************************
560 REM *   DOVOLJNA TACNOST ZA B-1 ITERACIJU  *
565 REM ****************************************
570 D=0:GOTO 520
575 REM ****************************************
580 REM *  TEST VREDNOST D2 POCINJE DA OSCILUJE *
585 REM ****************************************
590 D=1
595 J=K
600 GOTO 520
605 REM ****************************************
610 REM *      SVE ISTE VREDNOSTI U NIZU N     *
615 REM ****************************************
```

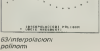
```
620 D=3
625 GOTO 595
630 REM ****************************************
635 REM *   IZDAVANJE REZULTATA I KOMENTARA    *
640 REM ****************************************
645 NI=3
650 GOSUB775
655 ON D+1 GOTO 660,675,695,735
660 PRINT"TACNOST DOSTIGNUTA"
665 GOSUB 795
670 RETURN
675 PRINT"TACNOST NIJE DOSTIGNUTA ZBOG GRESAKA
680 PRINT"ZAOKRUZIVANJA."
685 GOSUB 795
690 RETURN
695 PRINT"BROJ TACAKA MANJI OD 3, I ZBOG TOGA"
700 PRINT"NIJE BILO MOGUCE PROVERITI TACNOST "
705 PRINT
710 PRINT"ILI SE POMOCU OVIH VREDNOSTI"
715 PRINT"ZA X(I) I Y(I) TRAZENA TACNOST NE"
720 PRINT"MOZE DOSTICI (POVECATI BROJ TACAKA)"
725 GOSUB795
730 RETURN
735 PRINT"OTKRIVENE SVE ISTE VREDNOSTI"
740 PRINT"U NIZU X(I),"
745 PRINT
750 GOTO 380
755 REM
760 REM ****************************************
765 REM *      STAMPANJE NI PRAZNIH REDOVA     *
770 REM ****************************************
775 FOR M=1 TO NI:PRINT:NEXT M:RETURN
780 REM ****************************************
785 REM *  IZDAVANJE INTERPOLIRANE VREDNOSTI   *
790 REM ****************************************
795 PRINT
800 PRINT"INTERPOLIRANA VREDNOST FUNKCIJE"
805 PRINT"Y = ";Y:PRINT
810 RETURN
815 REM ****************************************
820 REM *  PREUREDJENJE ULAZNIH NIZOVA A I V   *
825 REM ****************************************
830 FOR I=1 TO B:N(I)=ABS(X-A(I)):NEXT I
835 FOR K=1 TO B
840 TE=1E38:I1=1
845 FOR I=1 TO B
850 IF N(I)>=TE THEN 860
855 TE=N(I):I1=I
860 NEXT I
865 N(I1)=1E38:X(K)=A(I1):V(K)=V(I1)
870 NEXT K
875 RETURN
880 END
```











gram na ulazu zahteva vrednost za željenu tačnost. U programu su navedena značenja promenljivih.

Greške koje se čine kada se interpolacija vrši na bazi polinoma, mogu se procenjivati korišćenjem formula za grešku koje su date u prethodnim nastavcima. Formula za grešku kod Newtonovog polinoma ista je kao i kod Lagranžovog. Ali da bi se dala i vizuelna ilustracija ponašanja interpolacionog polinoma, pogodno je dati grafički prikaz za tipične slučajeve.

Na slici 1 prikazan je interpolacioni polinom (svejedno koji, Lagranžov ili Njutnov jer oni uzimaju iste vrednosti) za kvadratnu funkciju. U ovom slučaju greška je nula jer je kvadratna funkcija specijalan slučaj polinoma pa se aproksimira bez greška. (Napominjemo da su unete vrednosti predstavljene gornjim levim temenom kvadratića.)

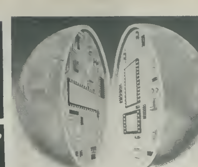
Na slici 2 je eksponencijalna funkcija koja nije polinom i kod koje se već [javlja] greška. Greška je veoma uočljiva na slici [3] gde je prikazana sinusna funkcija. Interpolacioni polinom znatno odstupa na krajevima intervala, što je tipično za ovu interpolaciju.

Na slici 4 je prikazan izmišljen eksperiment, tj. tačke su proizvoljno uzete. Tu se takođe vidi oscilovanje interpolacionog polinoma na krajevima intervala. Treba reći da pored polinomijalne interpolacije postoje i druge vrste interpolacija o kojima nije bilo reči. To su interpolacija racionalnom funkcijom, verižnim razlomkom, splajn interpolacionom funkcijom itd. Veoma često su ove druge vrste interpolacija pogodnija i daju bolje rezultate. Primera radi, na slici 5 su isti podaci sa slika 4 interpolirani splajn funkcijom (u poslednje vreme veoma popularan metod koji zaslužuje posebno razmatranje). Očigledno je da je aproksimacija „prirodnija".

Podvucimo još jednom da izbor interpolacionog metoda zavisi od tipa podataka na osnovu kojih se vrši interpolacija.

Mr Veljko Spasić





Računari iz mog ugla

KOMPJUTER U NAŠOJ MALOJ FIRMI

Dvehiljadite godine svako će imati svoj računski centar

Osnovni problem malih računara je u tome što su mali

Veliki privredni promašaji posmatraju se sa ekonomsko-političkog stanovišta a nikako da se shvati da su megalomanske investicija pre svega nečiji psihološki problem

Od megalomanije obično oboljevaju ambiciozni, a nesposobni rukovodioci Mučeni svojim ništavilom vrebaju priliku za velike role Pošto kompjuter ima pre svega statusnu vrednost a tek onda upotrebnu megalomanski direktor male firme će pokušati sebi da podigne značaj izgradnjom računskog centra Mali računari ne dolaze u obzir Sa malim se ne može biti zadovoljan čak ni ako vrši posao

Kada mala firma počne da zida računski centar to je jedan od pouzdanijih znakova da su stvari krenule loše Uostalom to ne mora biti nagoveštaj nego i uzrok propasti zajedno sa još nekom velikom investicijom i sve to zato što direktor ima potrebu da sazida spomenik sebi za života

Prodavci domaćih velikih računara svakako znaju za tu slabost nekih direktora — inače teško da bi uspeli nekom da uvale tako skupu robu Oni znaju kako mali direktor Perica zamišlja računski centar od računara se sve može saznati pritiskom na dugme Ponudiće mu na fini način kompjuter kao sredstvo kontrole svih onih koji pokušavaju da ga skinu Sva važna odluka u OOUR-ima moraće da se objedine pod njegovim mudrim rukovodstvom

Ni sam direktor nije svestan svoje nesvesti. On bi najviše voleo neki IBM makar polovan prastari model pa da može da prošeta poslovne prijatelje kroz 5—6 soba a oni da bleje u lampice što svetlucaju i magnetne trake koje se vrte

Naša surova realnost gde veliki računari ne mogu da se uvoze nateraće direktora da organizuje sastanak sa drugovima iz tvrtke DRP-DATA našeg poznatog proizvođača računara

Ovakvim sastancima treba da prisustvuju samo direktor i njegovi najbliži saradnici a nikako neki mlad inženjer koji bi mogao da pita zašto sve to košta 5—10 puta više nego u inostranstvu Iskusni ždarači poslovnih ručkova nikada neće postaviti tako neumesna pitanja Cena performanse sistema softver — to su sitnice koje mogu da obeshrabre drugove iz DRP-DATE, i dovedu u pitanje prijateljski poslovni ručak i uspešan nastavak dalje saradnje

Sledeći vrlo važan korak je progurati celu stvar kroz samoupravnu proceduru Srećom postoje oprobane metode Pitanje izgradnje računskog centra postaviće se na začelje dnevnog reda Prvo bi usledila duga i žučna diskusija o rasipanju spajalica i drugog kancelarijskog materijala Na kraju kada su svi već umorni i ispražnjeni, ustaje šef računovodstva koji sebe već vidi kao direktora računskog centra i predlaže investiciju od desetak milijardi On vrlo opširno obrazlaže kako će taj svemoćni računski centar u sadašnjoj složenoj situaciji rešiti sve naše probleme do 2000-te godine Pošto svi jedva čekaju da krenu kući niko neće ni registrovati ovakve budalaste tvrdnje smotanog šefa računovodstva, koji ne zna ništa ni o svojim trenutnim problemima

I tako računski centar ladno prolazi Dok o regresu za godišnji odmor zna svako nešto da kaže, te milijarde su potpuno van spoznaje normalnog čoveka I uopšte velika i mala glupost mnogo lakše prolaze nego srednje Srednja glupost može svako da vidi, dok ogromna i sićušne nije moguće sagledati bez posebnih pomagala Na nesreću, one su i najopasnije velike jer su velike a male jer ih mnogo ima

Pretpostavimo da je direktor male firme uspeo da završi računski centar i da firma nije potonula Računski centar neće povećati produktivnost jer će se isključivo koristiti za administrativne poslove Računar ne samo da neće smanjiti broj administrativnog osoblja, nego će ga čak i povećati Na primer, bivši šef računovodstva, a sada direktor računskog centra koji veze nema o računarima moraće da zaposli minimalno dvojicu pomoćnika koji nešto znaju Ako bi bio samo jedan ovaj bi ga ubrzo izgurao iz fotelje

Osim toga, razmnožavanje administracije obrnuto je srazmerno poslu koji treba obaviti Računski centar ma kako šugav i zastareo bio daleko premašuje potrebe male firme Da se računar ne bi dosađivao baš kao i osoblje koje je vrlo brzo izgustiralo onih par idiotskih igara koje se na njemu mogu igrati veštački se izmišlja posao I svi su onde pretrpani poslom koji sami sebi prave I nikako im nije jasno kako se nekada uopšte moglo raditi bez računskog centra

Iz izloženog se vidi da je *potpuno neprihvatljiva* svetska tendencija da se proizvode što manji, a što moćniji računari po niskim cenama Mi bismo morali razvijati za naše uslove daleko primereniju koncepciju — Price Without Power (u slobodnom prevodu um caruje snaga klade valja) Iako je na tom polju dosta učinjeno očito je da se stalo na pola puta Računski centar trebalo bi da preuzme funkciju koju je nekada imala crkva Trebalo bi da narodu uliva strahopoštovanje da bude što veći sa prozirnim zidovima da ljudi vide a da ne mogu da uđu Osoblje bi obavezno moralo da nosi bele mantile kao sve viđenije profesije (lekari arhitekte nuklearni fizičari maseri) Oprema bi morala da bude futurističkog dizajna sa što više svetlosnih efekata Šta ima unutra nije osobito važno — bitno je da ne bude prazno

Male računare bi trebalo zakonom zabraniti

Jelena Rupnik

Razbarušeni sprajtovi

Usijani džojstik

Stiglo nam je nekoliko saveta za „Elitu" odizvesnog Tamina iz Pančeva. Nakon skoka u hiper svemir potrebno je prići planeti, što često oduzima mnogo vremena. Da bi potrebno vreme bilo što kraće nemojte „gledati" u planetu, već u neki od preostala tri pravca.

Ako želite da prvim laserskim pucnjem oštetite udaljeni neprijateljski brod, potrebno je da ubacite projektil, dovedete neprijateljski brod na mušicu vašeg lasera i u trenutku kada začujete zvučni signal, zapucate.

Na gotovo svakoj trećoj stanici možete zaraditi veliku količinu novca kupovinom i prodajom trećeg od nazad proizvoda čija se vrednost menja pritiskom na taster 5 ili 6.

Toliko o Eliti. Marko Vukasović iz Zadra nam piše da je u igri „Commando" postigao skoro od 3075900 i da je „onaj ušljivi bunker" uništio 51 put. Smatra da je taj rezultat najbolji do sada postignut.

Ivan Stojković iz Niša javlja da je tako specijalizovao „Bouldardash II" da može čitavu igru da završi za 18 minuta (to mu je i rekord).

Cele iz Ljubljane nam kaže da je probao Mastertroniksove „budžetske" igre koje smo pominjali u prošlom broju i tvrdi da su sve osim „Spellodunda" izuzetno ružne, glupe i dosadne. Dobro, onda preporučujemo „Spellbound".

Ismet Arslanović iz Sarajeva kaže da je od nekog druga čuo da postoji spravica koja omogućava da budete praktično besmrtni u igrama i da se zove Game Killer. Znate šta to znači. Kad se ona koristi, sprečava se registrovanje preklapanja (itarekcije) sprajtova, pa zato ne samo da ne možete biti ubijani nego ne možete ni „ubiti". U nekim igrama kao što je Elite, ovo ne pali. Spravica košta 15 funti.

Milan Pantelić iz Beograda se žali da je zbog našeg saveta nabavio igru „Three Weeks in Paradise" i da je sada zaglavio negde usred igre i ne može dalje. CCCccc

Goran Kaporelić iz Beograda tvrdi da je igra „Flintstones" imbecilna i da mu je žao onih par hiljada koje je dao piratu za nju. Uz put tvrdi da je sagradio kuću za pola sata.

Za sam kraj evo jednog apela. Raštegorac Petar iz Novog Beograda moli da objavimo rešenje programa „Elite" za „Komodor" i završava pismo jednom izdvojenom rečenicom: VLASNICI COMMODORA VERUJU U DEJANA RISTANOVIĆA

BESMRTNOST ZA „KOMODOR 64"

Po učitavanju igre treba uneti odgovarajući poke, i tek zatim startovati igru.

BANDITS	POKE 4758 169
GYRUSS	POKE 10399,234 POKE 10400,234
HYPER OLYMPIC	POKE 14041 173
EVERYONE S WALLY	POKE 34481 157
DUMMY RUN	POKE 4306 165 POKE 4446 165
SPY HUNTER	POKE 15582 234 POKE 15583,234
QUASIMODO	POKE 13571 165
FRAK 64	POKE 22048 173
GROG S REVENGE	POKE 23606 173
BOOTY	POKE 21003 165
BEACH HEAD	POKE 8567 173 POKE 28486 173
	POKE 29765 173 POKE 34074 173
JET SET WILLY	POKE 14271 165
LAZY JONES	POKE 4251 173
ZAXXON	POKE 28400 173
BRUCE LEE	POKE 7462 165
TAPPER	POKE 15899 165
RAID OVER MOSCOW	POKE 4075 173 POKE 10057 173
	POKE 21668 173 POKE 30850,173 (životi)
	POKE 23429 173 POKE 24314 173 (diskovi)
FLIP AND FLOP 2	POKE 19342,234 POKE 19343 234
BUMPING BUGGIES	POKE 10705,173
KUNG FU MASTER	POKE 38649 165
SKOOL DAZE	POKE 7553 165 POKE 7623 165 (lines)
COMMANDO	POKE 2409 173 POKE 4854 173
ARABIAN NIGHTS	Posle učitavanja prvog dela programa treba uneti A=13190 POKE A, 165 POKE A+1 173
	POKE A+2 141 POKE A+3 165 POKE A+4,89
	POKE A+5 65
IMPOSSIBLE MISSION	Kada se učita loader snimljen standardnom brzinom, treba otkucati
	POKE 2305 141
	POKE 2306 19 POKE 2307,109
BLAGGER	POKE 3560 6
FORT APOCALYPSE	POKE 36338 153
FROGGER	POKE 22341 173
HARD HAT MACH	POKE 16577 173
HUNCHBACK	POKE 9521 234
	POKE 9522 234
	POKE 9523 234
MOON BUGGY	POKE 24161,173
NEPTUNE'S DAUGHTERS	POKE 7870 255
POOYAN	POKE 20834 173
SNOKIE	POKE 33242 55

BESMRTNOST ZA „SPEKTRUM"

Impossible mission

Učitajte bejzik deo sa MERGE i dodajte

15 POKE 45299 62 POKE 45300 53

startujte program sa RUN i nastavite učitavanje

Dynamite dan

Učitajte bejzik loader sa MERGE i u liniji 10 dodajte ispred RANDOMIZE USR POKE 51398,110 POKE 55755 0

Gyroscope

Premotajte traku iza uvodnog bejzika i naslovne slike i otkucajte sledeći program

```
10 FOR N-23296 TO 23331
20 READ A POKE N A NEXT N
30 RANDOMIZE USR 23296
40 DATA 49 40 9[illegible] 221 33 0 94 17 0
16[illegible] 62 [illegible]59 55 205 86 5 221 33 232 253
7 4 62 55 [illegible]5 205 86 5 175 50 16[illegible]
0 [illegible]95 94 206
```

Program startujte sa RUN i nastavite učitavanje

Commando

Postupak je isti kao i kod prethodnog programa tj ne treba učitavati bejzik loader i sliku

```
10 FOR N-23296 TO 23322
20 READ A PO E N,A NEXT N
30 RANDOMIZE USR 23296
40 DATA 49 128 91 221,33 188 91
128 16[illegible] 62 85 55 205,86 5 49 0 98
62 182 50 5 188,195 26 254
```

Gunfright

Traku premotajte iza bejzika i slike i otkucajte

```
10 LOAD   CODE LOAD   CODE
20 POKE 23457 201 RANDOMIZE USR [illegible]
30 POKE 41845 0 POKE 51049 [illegible]33 [illegible]
45250 0-POKE 53888,201 PRINT USR
24064
```

Tau ceti

Preskočite bejzik i sliku na traci otkucajte

```
10 FOR N-20480 TO 20506
20 READ A POKE N,A NEXT N
30 PRINT USR 20480
40 DATA 49 31,64 221 33 0 31 [illegible]
65 62 255 55 205,86,5 210,0 0 [illegible]
50 [illegible] 194 195 0 91
```

B C 'S quest for tires

Sve uraditi kao kod prethodnih igara, a program je

[illegible]



Hakerska priča

Sinopsis za vašu novu igru

Daš? Dam!

Cilj ove arkadne avanture je da uz pomoć 100 000 000 dolara, koje ste dobili od velike multinacionalne kompanije, omogućite sklapanje ugovora za izgradnju pet nuklearnih elektrana u našoj zemlji

Na početku igre treba da izaberete zemlju u kojoj ćete izvršiti svoju humanističku misiju Najbolje je da to bude neka zemlja koja već ima iskustva sa inostranim kreditima

Nakon što ste to uredili pred vama se pojavljuje ekran — plan grada na kome su obeležene sve značajnije ustanove Najpametnije je da se odmah uputite u gradsku većnicu, gde se održava koktel kome prisustvuju svi viđeniji ljudi

Na koktelu se igra poznata društvena igra — mrane fote Svetlo se naizmenično pali i gasi a vi treba, dok je mrak, da umuvate neprimetno izvesnu količinu dolara u džepove raznih ministara To nije lako, zato što se pored ministara oko vas muvaju i razni bezveznjaci pa ih u mraku možete pobrkati Kada se svetlo upali ministar će promeniti boju, on će pocrveneti ili pozeleneti, zavisno od toga da li ste mu uvalili dovoljno love ili ne U slučaju da je pozeleneo, moraćete ponovo da se jurite s njim po mraku, sve dok ne dobijete željenu nijansu Dakle na ovom nivou cilj je jednostavan ako ne i vrlo jednostavan treba da uvaljujete lovu ministrima sve dok ne pocrvene

Ako su vam pocrveneli ministri za energetiku, finansije zdravstvo, nauku i informisanje, kao i neki tipovi u uniformi pred vama se pojavljuje sledeći ekran banka u Švajcarskoj, gde ćete na svoj konto uložiti ostatak provizije i premiju koju ste dobili za uspešno obavljen posao Time ste već stekli rejting „dangerous"

Sa novom provizijom odlazite opet u istu zemlju, tj automatski se vraćate na prvi ekran na kome se nalazi plan grada Možete ponovo da odete na koktel u gradsku većnicu, ali, u svakoj narednoj poseti teže ćete postići da vam ministri pocrvene.

Pretpostavimo da vam nije uspelo da stignete sve ministre Ako ste, recimo, propustili da se izmirite sa ministrom za nauku i zdravstvo u sledećem nivou moraćete da obiđete što više instituta gde treba da skupite određen broj metra naučnih radova koji su razbacani po najskrovitijim mestima Međutim ne treba da uzimate bilo šta, jer ako greškom uzmete i one radove koji nisu pravilno orijentisani, moraćete da ih pojedete i tako im uništite svaki trag

Najnezgodnije je ako vam izmakne ministar za informisanje onda ćete morati da se rastrčite po redakcijama i da u svakoj kupite bar po jednog novinara Takođe ćete morati da posetite biblioteku i tamo uništite sve tekstove o štetnosti radijacije, da spaljujete sve publikacije u kojima nema dovoljno optimizma prema izgradnji nuklearnih elektrana itd.

Na poslednjem nivou, suočavate se sa javnošću putem televizijskih kamera U TV studiju treba biti naročito spretan i brz, pa na vreme izmaći stolice svim protivnicima i preseći mikrofonske kablove onima koji bi mogli biti protiv

Igra je završena kada na ovaj način uspete da jednoj zemlji uvalite 5 (pet) nuklearnih elektrana i izvučete živu glavu Tada se na ekranu pojavljuje veliki natpis „SVEGA IMA DA BUDE SAMO NAS NEMA DA BUDE", i vi automatski stičete rajting „deadly"

Jelena Rupnik

Top lista

Spektrum	Komodor	Amstrad
1 Movie	1 Kung Fu Master	1 Elite
2 Winter Games	2 Yie Ar Kung Fu	2 Sky Fox
3 Spellbound	3 Rock N Wrestle	3 Yie Ar Kung Fu
4 Barry McGuigen World Champions	4 Eidolon	4 Hypersports
5 Commando	5 Mercanary	5 They Sold a Milion
6 Yie Ar Kung Fu	6 Kane	6 Formula One Simulator
7 Rambo	7 Desert Fox	7 Caves Of Doom
8. Hypersports	8 Koronis Rift	8 Finders Keepers
9 Caves Of Doom	10 Winter Games	9 Who Dares Wins 2
10. Gunfright		10 Spellbound

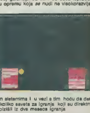
Igrajte „Elitu" bez varanja

PUCAJ, SINE!

Još jednom se (u istoriji) pokazalo da se od nepoštenja može sasvim lepo živeti Ovde se to odnosi na moj status u igri „Elita u kojoj sam tavorio sa dvestotinak kredita i malo opreme nekoliko prvih nedelja igranja Pošto mi je dozlogrdilo da me napucava svaka šuša koja doleti iz mraka odlučio sam da se privremeno odreknem časti i poštenja i na brzu ruku razbio celu flotu trgovačkih piratskih i policijskih brodova Švercovao sam narkotike oružje i robove, jurilo me je desetak lovaca na ucena ali su svi prošli lato (loše)

Sada mi je status, naravno, „Fugitive i ne verujem da ću ga skoro izmeniti Međutim, rejting mi je „Competent", a posada mi svaki čas viče „Right on, Commander!", pa verujem da ću lepo napredovati u službi

Još da napomenam da se nalazim u trećoj galaksiji da sam do sada isprobao svu opremu koja se nudi na visokorazvijenim sistemima i u vezi s tim hoću da dam nekoliko saveta za igranje koji su direktno proizišli iz dva meseca igranja



1 Ne bacajte pare na „Military laser I Prvo sam pojačao brod sa „Beam (zračnim) laserima, ali nisam bio zadovoljan Zatim sam kupio Minning' (rudarski) laser pa sam skupljao minerale i lepo napucavao okolo Kada sam sakupio dovoljno love nabavio sam taj vojni laser i utvrdio da guta mnogo energije, tako da sam morao da kupim jaču energetsku jedinicu
„Izuzetna probojna moć vojnog lasera je važila samo za male brodove ali se nijedan nije raspao bez bar tri pogotka Još gore sam prošao kada sam upucavao velike „Python' trgovačke brodove, koji nisu hteli da puknu ni posle deset pogodaka! Ovo je bilo previše jer sam potpuno isti efekat imao i sa „Beam' laserima, pa sam prodao vojni laser kupio ponovo rudarski koji am što isti osao sređuje sa tri-četiri pogotka am mi

omogućuje da skupljam minerala sa asteroida. Vrlo pozitivno na moral deluje kad vidite kako napadač (čitaj pirat) pogođen samo jedanput iskače kapsulom ostavljajući brod na milost i namilost vama!

2 Kupite „Docking computer". Tako ćete pobeći od gomile pirata kada ste već u sigurnosnoj zoni stanice a oni vas napadnu u velikom broju i uštedećete puno vremena koje ste do sada gubili na sletanje. Bez kompjutera u stanicu se ulazi dobro nacentrisan na ulaz bez rotiranja broda i pred sam ulaz sa maksimalnim gasom. Ovaj sistem retko kad omašuje osim kada loše nacentrirate ulaz.

3 Nađite dva blisko smeštena različita sistema, recimo siromašan poljoprivredni



i bogat industrijski i letite između njih sve dok ne sakupite dovoljno love za skok u novu galaksiju. Treća galaksija je vrlo interesantna.

4 Ako želite da igrate sa džojstikom, dovoljno je da pomerite palicu ulevo kada se na ekranu pojavi natpis „PRESS SPACE COMMANDER". Od sada će „ELITA" reagovati na svaki njegov pokret.

5 Snimajte svoj status u svakoj stanici. Da biste uštedeli trud ne startujte traku već snimite status u prazno. Ako kasnije budete nagđa napucani, to će vam biti početno mesto za novu igru. Tek kada vam se prispava snimite status stvarno na traku i isključite kompjuter.

P. S. Postao sam „Dangerous"!

Darko Stanojević

Spektrum



Comic Bakery

Smešna pekara

Ljudi iz redakcije „Računara" vole dobro da jedu. Svako ko je i jednom posetio redakciju zna da je to istina. Evo prave prilike da to i dokažemo. Ovo je prva igra koja se vrti oko jela još od prasterag Pecmana koje smo se dočepali. Igrali smo je jedno tri dana i uspeli da ispečemo gomile hleba (ali ne i da ga pojedemo). Priča je jednostavna. Vi ste pekar koga u procesu pečenja hleba ometaju rakuni (da, baš tako — rakuni) koji isključuju pećnice (da, baš tako — isključuju pećnice). Zato vi imate zračni pištolj koji smiruje rakune.

Grafike je ljupka, cilj je sladak, rakuni su simpatični, igra malo dosadi kada više niste gladni a nema ni slanih kifli. Ne izbegavajte ovu igru, ali je ne igrajte previše — osećate se kao da ste jeli previše testa. — Iromo



The Eidolon

Eidolon

Sad se svi pitaju šta je to Eidolon. Eidolon je mašina koja omogućuje putovanje kroz vreme i maštu. Da se ne biste previše opustili u putovanju kroz vreme i vašu maštu u hodnicima i pećinama kroz koje se putuje (da li te špilje predstavljaju maštu — mnogo je mračno) neprekidno nailazite na neverovatan izbor čudovišta. Čudovišta su najbolji deo ove igre. U njima ćete sigurno prepoznati pola svog komšiluka. Mnoga od čudovišta su pod uticajem jedne od sjajnih raznobojnih kugli koje se sreću u hodnicima. Crvene su opasne ali vam mogu pomoći protiv čudovišta. Zlatne povećavaju energiju vaše mašine. Plave menjaju tok vremena dok zelene transformišu jedno čudovište u drugo. Primetićete da se ne možete rešiti svih čudovišta na isti način. Svako od njih zahteva različit tretman. Kao u pravom životu, jel' te?

KOMODOR 64



Back to the future

Povratak u budućnost
Electric Dreams

Opet jedna igra rađena prema filmu. Ako ste ga gledali biće vam mnogo lakše da shvatite o čemu se radi. Glavni junak tj. vi vraćate se u prošlost. Vaš najveći problem je u tome što je vaša buduća keva više zainteresovana za vas nego za vašeg budućeg ćaleta. Treba da razrešite tu edipovsku situaciju inače vam se može desiti da uopšte ne postojite u budućnosti. Pored vaših dragih roditelja tu se muvaju još neki tipovi. Od toga je jedan naročito naprijatan stalno pokušava da vas nokautira i tako vas omela u nastojanju da organizujete susret svojih budućih roditelja sve u nadi da će se oni zavoleti mladi. Koliko ste u tome uspešni simbolično je predstavljeno fotografijama koje se nalaze u dnu ekrana. Prolazeći kroz razne prostorije treba da pokupite pet različitih predmeta. Možete ih upotrebiti da vidite kakav efekat imaju na druge likove. Mada je grafika dosta dobro urađena, pametnije je da odete u bioskop i vidite kako to stvarno izgleda.



...alex

Leisure Genius

Još jedna igrica namenjena potencijalnim vozačima formule jedan. Pruža vam se mogućnost da birate između 17 svetskih „Gran pri" staza a ukoliko ni jedna nije po vašem ukusu možete i sami da dizajnirate stazu a zatim je snimite na kasetofon ili disk. Igru mogu igrati dva igrača jedan protiv drugog. Menjanje brzine je automatsko s tim što je maksimalna brzina 240 km/h. Na dnu ekrana možete videti svoj trenutni položaj na stazi, broj krugova koje treba preći kao i brzinu kojom se krećete. Sva mudrost je otprilike u tome da ne uletete pod punim gasom u krivine jer vam se može desiti neprijatna stvar da izletite sa staze što će vam drastično smanjiti brzinu a vašem protivniku omogućiti da vas prestigne za tren oka. Grafički ugođaj je prilično dobar mada nedostaju raznovrsni detalji pored staze. Prateći zvuci prilikom menjanja brzina kao i škripanje guma u krivinama pomoći će vam da se maksimalno uživite u ulogu vozača formule jedan. Ako ste uživali igrajući Pitstop I i II, Pole Position, Revs i sl. pod hitno nabavite i ovu igru da biste upotpunili svoju kolekciju.

